

# **Instituto Politécnico Nacional**

Escuela Superior de Cómputo

Análisis y Diseño de Sistemas

Grupo: 5CM3

## **Propuestas de requerimientos no funcionales para el desarrollo de un sistema de gestión escolar (SAES)**

Alumno: Hervert Martínez Jairo Jesús

Docente: M. Maldonado Castillo Idalia

*Fecha de Entrega: 26 de Septiembre 2024*

## **CONTENTS**

<b>1</b>	<b>Introducción</b>	<b>2</b>
<b>2</b>	<b>Objetivo</b>	<b>2</b>
<b>3</b>	<b>Principales tipos de requerimientos no funcionales</b>	<b>2</b>
<b>4</b>	<b>Importancia de los Requerimientos No Funcionales</b>	<b>3</b>
<b>5</b>	<b>Ejemplos de requerimientos no funcionales</b>	<b>4</b>

# Requerimientos No Funcionales

HERVERT MARTÍNEZ JAIRO JESÚS

## 1. INTRODUCCIÓN

Los requerimientos no funcionales en el desarrollo de software se refieren a los criterios que definen el funcionamiento del sistema de manera indirecta, es decir, cómo se comportará el software en términos de rendimiento, seguridad, usabilidad, entre otros, en lugar de lo que el software hará específicamente (como sucede con los requerimientos funcionales). Estos son esenciales porque influyen en la experiencia del usuario, la escalabilidad del sistema y su sostenibilidad a largo plazo.

## 2. OBJETIVO

El objetivo de esta tarea es proponer un conjunto de requerimientos no funcionales clave para el desarrollo de un sistema de gestión escolar (SAES). A través de este análisis, se busca garantizar que el sistema cumpla con estándares de rendimiento, seguridad, escalabilidad y usabilidad, entre otros aspectos esenciales, para asegurar una experiencia de usuario eficiente y confiable. Además, se pretende identificar las características necesarias para la sostenibilidad y el mantenimiento a largo plazo del sistema.

## 3. PRINCIPALES TIPOS DE REQUERIMIENTOS NO FUNCIONALES

- **Rendimiento (Performance):**
  - Relacionado con la capacidad del sistema para responder y procesar tareas en un tiempo determinado. Involucra tiempos de respuesta, latencia, velocidad de procesamiento, uso de memoria y capacidad de manejar múltiples usuarios o transacciones simultáneamente.
  - **Ejemplo:** "El sistema debe poder procesar 10,000 transacciones por minuto."
- **Escalabilidad:**
  - Mide la capacidad del sistema para crecer y manejar un volumen creciente de trabajo sin perder eficiencia.
  - **Ejemplo:** "La plataforma debe poder aumentar su capacidad para soportar un 50% más de usuarios durante los picos de tráfico."
- **Seguridad:**
  - Define cómo el sistema debe protegerse contra accesos no autorizados, violaciones de datos y ciberataques. Incluye aspectos como autenticación, autorización, encriptación de datos, auditoría y manejo de vulnerabilidades.
  - **Ejemplo:** "El sistema debe encriptar todas las contraseñas utilizando algoritmos de encriptación de 256 bits."
- **Confiabilidad (Reliability):**
  - Mide la capacidad del sistema para funcionar correctamente sin fallar durante un período de tiempo específico. Involucra conceptos como tiempo de inactividad (*downtime*), tasa de fallos y tolerancia a fallos.

- **Ejemplo:** "El sistema debe estar disponible el 99.9% del tiempo."
- **Usabilidad (Usability):**
  - Evalúa qué tan fácil es para los usuarios interactuar con el sistema. Involucra aspectos como la facilidad de aprendizaje, la eficiencia en la realización de tareas, la accesibilidad y la satisfacción del usuario.
  - **Ejemplo:** "El sistema debe permitir a los usuarios completar una operación de búsqueda en menos de 3 clics."
- **Mantenibilidad (Maintainability):**
  - Se refiere a la facilidad con la que el software puede ser corregido, actualizado o adaptado a nuevas necesidades. Implica la claridad del código, modularidad, documentación, y la capacidad de aplicar parches o mejoras sin interrumpir el servicio.
  - **Ejemplo:** "Las actualizaciones del sistema deben poder aplicarse sin más de 1 hora de inactividad."
- **Portabilidad (Portability):**
  - Indica cuán fácilmente el software puede ser trasladado a diferentes entornos o plataformas. Puede referirse a la capacidad de ejecutarse en distintos sistemas operativos, navegadores o hardware.
  - **Ejemplo:** "La aplicación debe funcionar en las últimas tres versiones de los navegadores principales (Chrome, Firefox, Safari)."
- **Compatibilidad (Compatibility):**
  - Evalúa cómo el sistema interactúa con otros sistemas, aplicaciones o versiones anteriores. Se puede referir a la interoperabilidad entre sistemas o la compatibilidad hacia atrás (*backward compatibility*).
  - **Ejemplo:** "El sistema debe ser compatible con la versión anterior de la base de datos."
- **Disponibilidad (Availability):**
  - Relacionado con el tiempo que el sistema está operativo y accesible para los usuarios. Es un factor crítico en sistemas que necesitan estar en línea constantemente.
  - **Ejemplo:** "El servicio debe estar disponible las 24 horas del día, los 7 días de la semana."
- **Recuperación ante desastres (Disaster Recovery):**
  - Se refiere a la capacidad del sistema para recuperarse de fallos o desastres, garantizando la restauración de datos y la continuidad del servicio.
  - **Ejemplo:** "El sistema debe poder recuperarse de una interrupción mayor en menos de 30 minutos sin pérdida de datos."

#### 4. IMPORTANCIA DE LOS REQUERIMIENTOS NO FUNCIONALES

Aunque los requerimientos funcionales (lo que el sistema debe hacer) suelen ser los más visibles para los usuarios, los no funcionales son críticos para garantizar que el sistema no solo funcione, sino que lo haga de manera eficiente, segura y sostenible. Si no se cumplen los requerimientos no funcionales, la experiencia del usuario puede verse

afectada negativamente, incluso si todas las funcionalidades previstas están presentes y son correctas.

Por ejemplo, una aplicación de banca en línea podría cumplir perfectamente con todas sus funcionalidades (transferencias, pagos, consultas de saldo, etc.), pero si no es segura o tiene tiempos de respuesta lentos, perderá la confianza y satisfacción de sus usuarios.

## **5. EJEMPLOS DE REQUERIMIENTOS NO FUNCIONALES**

En esta sección se muestran las propuestas de requerimientos no funcionales planteadas durante la clase de análisis y diseño de sistemas, los cuales son requerimientos para un sistema de gestión escolar como lo es el SAES.

1. El sistema deberá realizar el cifrado de todos los datos sensibles utilizando algoritmos de encriptación de 256 bits para garantizar la seguridad.
2. El sistema deberá contar con un diseño responsivo, adaptándose automáticamente a dispositivos móviles, tabletas y computadoras de escritorio, garantizando una experiencia de usuario coherente.
3. El tiempo de respuesta del sistema deberá ser menor a 2 segundos al procesar solicitudes de hasta 500 usuarios concurrentes, asegurando un rendimiento óptimo.
4. El sistema deberá soportar incrementos en la carga de usuarios hasta en un 100% durante los periodos pico sin comprometer su rendimiento.
5. La interfaz del sistema deberá ser intuitiva y fácil de usar, con una curva de aprendizaje inferior a 30 minutos para los usuarios nuevos.
6. Las actualizaciones del sistema deberán poder aplicarse con un tiempo de inactividad máximo de 1 hora, garantizando la mantenibilidad y continuidad del servicio.
7. El sistema deberá ser compatible con las últimas tres versiones de los principales navegadores (Chrome, Firefox y Safari) para asegurar la portabilidad.
8. La disponibilidad del sistema deberá ser del 99.9% durante el año, permitiendo un tiempo de inactividad máximo de 8.76 horas anuales.
9. Los respaldos de la base de datos deberán realizarse de manera automática cada 24 horas, asegurando la recuperación de datos ante fallos.
10. En caso de una interrupción mayor, el sistema deberá garantizar la restauración completa en menos de 30 minutos, manteniendo la continuidad operativa y la integridad de los datos.