

# **Requisitos No Funcionales**

# RNF

Los **requisitos no funcionales** se refieren a los criterios que se pueden usar para juzgar el funcionamiento de un sistema, en particular los aspectos que no están relacionados con los comportamientos específicos de las funciones del sistema.

Estos requisitos son fundamentales para garantizar la calidad, usabilidad, escalabilidad, rendimiento y seguridad del sistema, entre otros aspectos. El proceso de manejo de requisitos no funcionales incluye su identificación y análisis, especificación, verificación y validación, y gestión de cambios.

# Identificación y Análisis de RNF

## 1. Identificación:

- **Entrevistas con Partes Interesadas:** Incluir discusiones específicas sobre expectativas de rendimiento, seguridad, mantenimiento, y más, durante las entrevistas con los usuarios y otros interesados.
- **Análisis de Sistemas Similares:** Revisar cómo sistemas comparables manejan aspectos no funcionales puede proveer insights valiosos.
- **Revisión de Estándares y Regulaciones:** Identificar requisitos no funcionales necesarios para cumplir con cumplimientos legales y estándares de la industria.

## 2. Análisis:

- **Priorización:** Determinar qué requisitos no funcionales son más críticos en base a los riesgos y el impacto potencial en el negocio y en los usuarios.
- **Interdependencias:** Analizar cómo los requisitos no funcionales impactan o están conectados con los requisitos funcionales y otros requisitos no funcionales.

# Especificación de RNF

## 1. Documentación clara:

- Los requisitos no funcionales deben ser específicos, medibles y realistas. Por ejemplo, "El sistema debe cargar cada página en menos de 3 segundos bajo una carga de usuario estándar".

## 2. Uso de herramientas de especificación:

- **Listas de comprobación:** Para garantizar que todas las áreas claves como la seguridad, la escalabilidad y la accesibilidad se cubren adecuadamente.
- **Modelos de calidad como ISO/IEC 25010:** Para estructurar los requisitos no funcionales en categorías como mantenibilidad, usabilidad, fiabilidad, rendimiento, y seguridad.

# Verificación y Validación de RNF

## 1. Verificación:

- **Revisiones de Diseño:** Determinar si la arquitectura propuesta y las decisiones de diseño satisfacen los requisitos no funcionales especificados.
- **Inspecciones y Revisiones de Código:** Verificar aspectos como la eficiencia del código y la seguridad a través de análisis estático y revisiones de pares.

## 2. Validación:

- **Pruebas de Rendimiento:** Ejecutar pruebas de carga y estrés para validar el rendimiento y la escalabilidad del sistema.
- **Auditorías de Seguridad:** Realizar pruebas de penetración y otras técnicas de auditoría de seguridad para validar los requisitos de seguridad.
- **Evaluaciones de Usabilidad:** Realizar pruebas de usabilidad con usuarios reales para asegurar que el sistema cumple con los estándares de usabilidad.

# Gestión de Cambios en los RNF

## 1. Proceso de Gestión de Cambios:

- Establecer un claro proceso de revisión y aprobación para cualquier cambio en los requisitos no funcionales, asegurando que todos los cambios sean analizados en términos de impacto y viabilidad.

## 2. Herramientas de seguimiento:

- Implementar sistemas de seguimiento de requerimientos que registren no solo los cambios en los requisitos funcionales, sino también los aspectos no funcionales, dando visibilidad a cómo estos cambios podrían afectar otras áreas del sistema.

## 3. Comunicación constante:

- Fomentar una comunicación fluida y constante entre los desarrolladores, diseñadores, arquitectos de sistema y partes interesadas para garantizar que todos los cambios se comprenden y se gestionan de manera efectiva.

# Conclusión

La gestión efectiva de los requisitos no funcionales es crucial para el éxito del proyecto, ya que impactan directamente en la experiencia del usuario y en la calidad percibida del sistema.

Su manejo adecuado requiere un enfoque metódico y detallado desde la identificación hasta la gestión de cambios, asegurándose de que estos requisitos sean bien entendidos, implementados correctamente y continuamente ajustados según sea necesario durante el ciclo de vida del desarrollo del software.