

# Contents

## Clase 4: Documentación de Requisitos de Calidad: Mejores Prácticas y Herramientas

### 1. Objetivos de la Clase:

- Comprender la importancia de una documentación clara y completa de los requisitos de calidad.
- Conocer las mejores prácticas para documentar los requisitos de calidad, incluyendo el uso de plantillas y estándares.
- Explorar diferentes herramientas y técnicas para la gestión y documentación de requisitos.
- Aprender a crear casos de prueba basados en la documentación de requisitos.

### 2. Contenido Teórico Detallado:

- **La Importancia de la Documentación de Requisitos de Calidad:**
  - **Comunicación efectiva:** La documentación sirve como un canal de comunicación claro y preciso entre todos los stakeholders (clientes, desarrolladores, testers, etc.). Evita ambigüedades y malas interpretaciones.
  - **Trazabilidad:** Permite rastrear cada requisito a lo largo del ciclo de vida del desarrollo, desde su origen hasta su implementación y prueba. Facilita la gestión de cambios y la identificación de impactos.
  - **Base para las pruebas:** Sirve como la base para la creación de casos de prueba, garantizando que todos los requisitos sean verificados y validados.
  - **Gestión de cambios:** Facilita la gestión de cambios en los requisitos, permitiendo evaluar el impacto de las modificaciones en el sistema.
  - **Conocimiento organizacional:** Preserva el conocimiento sobre los requisitos, incluso si los miembros del equipo cambian.
- **Mejores Prácticas para la Documentación de Requisitos:**
  - **Claridad y Precisión:** Utilizar un lenguaje claro, conciso y sin ambigüedades. Evitar jerga técnica innecesaria.
  - **Especificidad:** Cada requisito debe ser específico y medible. Evitar términos vagos como "rápido" o "fácil de usar". En su lugar, usar "La aplicación debe cargar la página principal en menos de 2 segundos" o "El usuario debe poder completar la tarea en menos de 3 clics".
  - **Compleitud:** Asegurarse de que todos los aspectos relevantes del requisito estén documentados. Incluir información sobre la prioridad, el origen, los criterios de aceptación y las dependencias con otros requisitos.
  - **Consistencia:** Los requisitos deben ser consistentes entre sí y con la arquitectura y el diseño del sistema. Evitar contradicciones y conflictos.
  - **Trazabilidad:** Establecer enlaces explícitos entre los requisitos y otros artefactos del desarrollo, como casos de uso, modelos de datos, diseño de interfaces y casos de prueba.
- **Plantillas y Estándares para la Documentación de Requisitos:**
  - **Plantillas:** Utilizar plantillas estandarizadas para la documentación de requisitos. Estas plantillas suelen incluir campos para el ID del requisito, el nombre, la descripción, la prioridad, el origen, los criterios de aceptación y el estado. Ejemplos de plantillas incluyen las basadas en casos de uso o historias de usuario.
  - **Estándares:** Seguir estándares de la industria para la documentación de requisitos, como IEEE 830 (Recommended Practice for Software Requirements Specifications). Estos estándares proporcionan directrices sobre la estructura y el contenido de la documentación de requisitos.
- **Herramientas para la Gestión y Documentación de Requisitos:**
  - **Herramientas de gestión de requisitos:** Estas herramientas facilitan la captura, el seguimiento, la gestión de cambios y la trazabilidad de los requisitos. Ejemplos: Jira, Azure DevOps, Polarion ALM, Jama Connect, ReQtest.
  - **Hojas de cálculo:** Pueden ser útiles para proyectos pequeños o para empezar, pero no ofrecen la escalabilidad ni la funcionalidad de las herramientas especializadas.
  - **Procesadores de texto:** Similares a las hojas de cálculo, útiles para empezar, pero limitados en funcionalidades avanzadas de gestión.

- **Documentación de Requisitos No Funcionales (Ejemplos):**

- **Rendimiento:**

- \* *Ejemplo:* "El sistema debe ser capaz de manejar 1000 usuarios concurrentes sin degradar el tiempo de respuesta en más de un 10%."
    - \* *Documentación adicional:* Especificar el entorno de prueba, las métricas de rendimiento (tiempo de respuesta, throughput, utilización de recursos), y los criterios de aceptación.

- **Seguridad:**

- \* *Ejemplo:* "El sistema debe proteger la información personal del usuario mediante encriptación AES-256."
    - \* *Documentación adicional:* Especificar los algoritmos de encriptación, los protocolos de autenticación y autorización, y los controles de acceso.

- **Usabilidad:**

- \* *Ejemplo:* "El usuario debe ser capaz de completar el proceso de pago en menos de 5 pasos."
    - \* *Documentación adicional:* Especificar los flujos de usuario, el diseño de la interfaz, los criterios de evaluación de la usabilidad (ej: System Usability Scale - SUS), y los roles de usuario.

### 3. Ejemplos y Casos de Estudio:

- **Caso de Estudio: Documentación de Requisitos para una Aplicación de Comercio Electrónico**

- **Requisito Funcional:** "El sistema debe permitir a los usuarios buscar productos por nombre, categoría y precio."

- \* *Documentación adicional:* Especificar los campos de búsqueda, los operadores de búsqueda (AND, OR, NOT), los criterios de ordenación de los resultados, y el número de resultados por página.

- **Requisito No Funcional (Rendimiento):** "El sistema debe mostrar los resultados de la búsqueda en menos de 3 segundos."

- \* *Documentación adicional:* Especificar el volumen de datos en la base de datos, el número de usuarios concurrentes, y el hardware del servidor.

- **Requisito No Funcional (Seguridad):** "El sistema debe proteger la información de las tarjetas de crédito de los usuarios mediante encriptación PCI DSS."

- \* *Documentación adicional:* Especificar los estándares de encriptación, los controles de acceso, y las auditorías de seguridad.

### 4. Problemas Prácticos y Ejercicios con Soluciones:

- **Ejercicio 1:**

- **Problema:** Escribe tres requisitos de calidad (uno funcional y dos no funcionales) para un sistema de gestión de biblioteca. Utiliza una plantilla estandarizada para la documentación.

- **Solución (Ejemplo):**

- \* *ID Requisito:* LIB-RF-001
    - \* *Nombre Requisito:* Búsqueda de libros
    - \* *Descripción:* El sistema debe permitir a los usuarios buscar libros por título, autor y ISBN.
    - \* *Prioridad:* Alta
    - \* *Origen:* Requisito del usuario
    - \* *Criterios de Aceptación:* El sistema muestra una lista de libros que coinciden con los criterios de búsqueda en un tiempo máximo de 2 segundos.
    - \* *ID Requisito:* LIB-NF-001
    - \* *Nombre Requisito:* Tiempo de respuesta

- \* *Descripción:* El tiempo de respuesta para cualquier operación del sistema no debe exceder los 3 segundos.
- \* *Prioridad:* Alta
- \* *Origen:* Requisito del sistema
- \* *Criterios de Aceptación:* El tiempo de respuesta se mide utilizando una herramienta de monitorización de rendimiento.
- \* *ID Requisito:* LIB-NF-002
- \* *Nombre Requisito:* Seguridad de la información
- \* *Descripción:* La información de los usuarios (datos personales, historial de préstamos) debe ser protegida mediante encriptación y controles de acceso.
- \* *Prioridad:* Alta
- \* *Origen:* Requisito legal
- \* *Criterios de Aceptación:* Se realizan pruebas de penetración para verificar la seguridad del sistema.

- **Ejercicio 2:**

- **Problema:** Dada la descripción de un sistema de reservas de vuelos online, identifica los requisitos de calidad (funcionales y no funcionales) y priorízalos utilizando la técnica MoSCoW. Documenta al menos 5 requisitos.
- **Solución:** (La solución variará según la descripción del sistema de reservas de vuelos proporcionada. La clave es identificar los requisitos y luego clasificarlos como *Must have*, *Should have*, *Could have* o *Won't have this time*).

## 5. Materiales Complementarios Recomendados:

- **Libros:**
  - "Software Requirements" de Karl Wieggers and Joy Beatty
  - "Writing Effective Use Cases" de Alistair Cockburn
  - "Requirements Engineering: From System Goals to UML Models to Software Specifications" de Axel van Lamsweerde
- **Artículos y Documentos:**
  - IEEE 830 - Recommended Practice for Software Requirements Specifications
  - Documentación de herramientas de gestión de requisitos (Jira, Azure DevOps, etc.)
- **Cursos Online:**
  - Cursos de Requirements Engineering en plataformas como Coursera, Udemy, edX.