

# Contents

## Clase 6: Profundizando en Requisitos de Usabilidad y Confiabilidad

### 1. Objetivos de la Clase:

- Comprender en profundidad los requisitos de usabilidad y su impacto en la experiencia del usuario.
- Aprender a definir y medir los requisitos de usabilidad de manera efectiva.
- Analizar la importancia de la confiabilidad y las métricas clave para su evaluación.
- Diseñar estrategias para asegurar la confiabilidad del software.

### 2. Contenido Teórico Detallado:

#### • Usabilidad:

- **Definición Extendida:** La usabilidad se refiere a la facilidad con la que los usuarios pueden aprender a usar un producto de software, la eficiencia con la que pueden realizar tareas, la satisfacción que obtienen al usarlo y la propensión a errores.
- **Principios de Usabilidad (Nielsen):**
  - \* Visibilidad del estado del sistema: Mantener a los usuarios informados sobre lo que está sucediendo.
  - \* Coincidencia entre el sistema y el mundo real: Hablar el lenguaje del usuario.
  - \* Control y libertad del usuario: Ofrecer "salidas de emergencia" claras.
  - \* Consistencia y estándares: Seguir convenciones y estándares de la industria.
  - \* Prevención de errores: Diseñar para evitar errores y ofrecer recuperación.
  - \* Reconocimiento en lugar de recuerdo: Minimizar la carga cognitiva.
  - \* Flexibilidad y eficiencia de uso: Permitir personalización y atajos.
  - \* Diseño estético y minimalista: Evitar información irrelevante.
  - \* Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de los errores: Mensajes de error claros y soluciones.
  - \* Ayuda y documentación: Proporcionar información fácil de buscar y comprender.
- **Métricas de Usabilidad:**
  - \* Tiempo de aprendizaje: ¿Cuánto tiempo tarda un usuario en aprender a usar el sistema?
  - \* Eficiencia: ¿Cuánto tiempo tarda un usuario en completar una tarea específica?
  - \* Tasa de errores: ¿Cuántos errores cometen los usuarios al intentar completar una tarea?
  - \* Tasa de éxito: ¿Qué porcentaje de usuarios puede completar una tarea con éxito?
  - \* Satisfacción del usuario: ¿Qué tan satisfechos están los usuarios con el sistema? (medido a través de encuestas como SUS - System Usability Scale).
- **Técnicas de Evaluación de Usabilidad:**
  - \* Pruebas con usuarios: Observar a los usuarios reales interactuar con el software.
  - \* Inspección de expertos (Heurística): Expertos en usabilidad evalúan el software basándose en heurísticas conocidas.
  - \* Walkthroughs cognitivos: Simulación del proceso mental de un usuario al interactuar con el software.
  - \* Encuestas y cuestionarios: Recopilar opiniones de los usuarios sobre la usabilidad del software.
- **Documentación de Requisitos de Usabilidad:** Los requisitos deben ser específicos, medibles, alcanzables, relevantes y con plazos definidos (SMART). Ejemplo: "El usuario debe poder completar el proceso de pago en menos de 3 minutos en el 95% de los casos".

#### • Confiabilidad:

- **Definición Extendida:** La confiabilidad es la probabilidad de que un sistema funcione correctamente durante un período de tiempo específico, en un entorno determinado. No solo se trata de la ausencia de fallos, sino también de la capacidad de recuperarse rápidamente de ellos.
- **Métricas de Confiabilidad:**
  - \* MTBF (Mean Time Between Failures): Tiempo medio entre fallos. Indica la frecuencia con la que falla un sistema.

- \* MTTR (Mean Time To Repair): Tiempo medio de reparación. Indica cuánto tiempo tarda en repararse un sistema después de un fallo.
- \* Disponibilidad: Porcentaje de tiempo que un sistema está operativo. Se calcula como  $MTBF / (MTBF + MTTR)$ .
- \* Tasa de Fallos (Failure Rate): Número de fallos que ocurren en un período de tiempo específico.
- **Estrategias para Mejorar la Confiabilidad:**
  - \* Diseño robusto: Diseñar el software para que sea tolerante a fallos.
  - \* Pruebas exhaustivas: Realizar pruebas exhaustivas para identificar y corregir errores.
  - \* Redundancia: Implementar sistemas redundantes para que, si un componente falla, otro pueda tomar su lugar.
  - \* Monitoreo continuo: Monitorear el sistema para detectar y corregir problemas antes de que causen fallos.
  - \* Gestión de errores: Implementar mecanismos para detectar, informar y recuperarse de los errores.
  - \* Planificación de la recuperación ante desastres: Tener un plan para restaurar el sistema en caso de un desastre.
- **Documentación de Requisitos de Confiabilidad:** Los requisitos deben especificar el nivel de confiabilidad deseado. Ejemplo: "El sistema debe tener una disponibilidad del 99.99%".

### 3. Ejemplos y Casos de Estudio:

- **Caso de Estudio: Aplicación de Reserva de Vuelos:**
  - **Requisitos de Usabilidad:**
    - \* El usuario debe poder encontrar vuelos que cumplan con sus criterios (destino, fechas, precio) en menos de 15 segundos.
    - \* El usuario debe poder completar el proceso de reserva en menos de 5 minutos.
    - \* El usuario debe poder entender fácilmente los términos y condiciones de la reserva.
    - \* La tasa de errores al ingresar la información de pago debe ser inferior al 2%.
  - **Requisitos de Confiabilidad:**
    - \* El sistema debe tener una disponibilidad del 99.9%.
    - \* El tiempo medio entre fallos (MTBF) debe ser de al menos 1000 horas.
    - \* El tiempo medio de reparación (MTTR) no debe exceder 1 hora.
- **Ejemplo de la vida real: El fallo de Therac-25 (mencionado en la clase 1).** Este caso sirve para ilustrar las consecuencias extremas de no tener en cuenta los requisitos de confiabilidad y usabilidad en sistemas críticos.

### 4. Problemas Prácticos y Ejercicios:

1. **Ejercicio de Usabilidad:** Diseña una interfaz de usuario para una aplicación móvil de gestión de contraseñas. Identifica los principios de usabilidad clave que aplicaste y cómo los implementaste. ¿Cómo medirías la usabilidad de tu diseño?
  - **Solución:** (La solución variará según el diseño, pero debe incluir la aplicación de principios de Nielsen, consideración de métricas como tiempo de aprendizaje, eficiencia y tasa de errores, y la justificación de las decisiones de diseño).
2. **Ejercicio de Confiabilidad:** Considera un sistema de comercio electrónico. ¿Qué estrategias implementarías para asegurar que el sistema sea confiable y esté disponible incluso durante picos de tráfico (por ejemplo, durante el Black Friday)? ¿Qué métricas utilizarías para monitorear la confiabilidad del sistema?
  - **Solución:** (La solución debe incluir estrategias como redundancia de servidores, balanceo de carga, caching, optimización de la base de datos, monitoreo continuo, planificación de la recuperación ante desastres y el uso de métricas como disponibilidad, MTBF, MTTR, y tasa de fallos).
3. **Análisis de un Sitio Web:** Elige un sitio web popular. Evalúa su usabilidad y confiabilidad. ¿Qué aspectos podrían mejorarse? Justifica tus respuestas basándote en los conceptos aprendidos en clase.
  - **Solución:** (La solución dependerá del sitio web elegido, pero debe aplicar los conceptos teóricos)

de usabilidad y confiabilidad y proporcionar ejemplos concretos).

## **5. Materiales Complementarios Recomendados:**

- **Libros:**
  - "Don't Make Me Think: A Common Sense Approach to Web Usability" de Steve Krug.
  - "The Design of Everyday Things" de Don Norman.
  - "Software Reliability Engineering" de John Musa.
- **Artículos:**
  - Artículos de Jakob Nielsen sobre usabilidad (Nielsen Norman Group).
  - Investigaciones sobre métricas de confiabilidad de software.
- **Cursos online:**
  - Cursos de Coursera o edX sobre usabilidad y diseño de experiencia de usuario (UX).
  - Cursos sobre ingeniería de confiabilidad de software.
- **Estándares:**
  - ISO 25010 (Características de Calidad del Producto Software): Define modelos de calidad incluyendo usabilidad y confiabilidad.