Fundamentos de Ingeniería de Software: Análisis, Metodologías y Ciclo de Vida

Trabajo Práctico en clase - Modelos y Sistemas Escuela Técnica N°1 de Muñiz  
7° 3º Año de Secundaria   
Profesor: Omar Alejandro Bazar  
Alumno: Brandon rivas

Tema 1: Análisis Funcional y Requerimientos del Software

1. Explica qué es el análisis funcional en el contexto del desarrollo de software y por qué es importante.

2. Define qué son los requerimientos funcionales y no funcionales. Proporciona un ejemplo de cada uno.

3. Describe el proceso de elicitación de requerimientos. ¿Qué técnicas se pueden utilizar para obtener información de los stakeholders?

4. ¿Qué es un caso de uso y cómo se utiliza en el análisis funcional? Proporciona un ejemplo sencillo.

5. Explica la importancia de la priorización de requerimientos en un proyecto de software. ¿Qué factores se deben considerar al priorizar? Tema 2: Metodologías Ágiles

6. Define qué son las metodologías ágiles y cómo se diferencian de las metodologías tradicionales de desarrollo de software.

7. Enumera y describe brevemente los cuatro valores fundamentales del Manifiesto Ágil.

8. Explica qué es Scrum y describe tres de sus elementos principales (por ejemplo, Sprint, Product Backlog, Scrum Master).

9. ¿Qué es un "sprint" en el contexto de las metodologías ágiles y cuál es su propósito?

10. Describe qué es una "historia de usuario" y cómo se utiliza en el desarrollo ágil de software. Tema 3: Ciclo de Vida del Software

11. Define qué es el ciclo de vida del software y enumera sus fases principales.

12. Explica la diferencia entre el modelo en cascada y el modelo iterativo de desarrollo de software.

13. Describe la fase de pruebas en el ciclo de vida del software. ¿Por qué es crucial esta fase?

14. ¿Qué es el mantenimiento del software y por qué es importante considerarlo como parte del ciclo de vida?

15. Explica qué es la "fase de análisis de requisitos" en el ciclo de vida del software y por qué es fundamental para el éxito del proyecto.  
  
Resolución:   
  
Tema 1: Análisis Funcional

¿Qué es el análisis funcional en el desarrollo de software y por qué es importante?  
 Es el proceso de descomponer un sistema en sus componentes funcionales, identificando qué debe hacer el software para cumplir con los objetivos del usuario y los requisitos del negocio. Es importante porque permite entender claramente las expectativas de los stakeholders y asegura que el sistema cumpla con los objetivos previstos.

Requerimientos funcionales y no funcionales:

Requerimientos funcionales: Describen qué debe hacer el sistema (funciones específicas).  
 Ejemplo: "El sistema debe permitir a los usuarios registrarse con una dirección de correo electrónico y contraseña".

Requerimientos no funcionales: Detallan cómo el sistema debe comportarse (calidad, rendimiento, seguridad).  
 Ejemplo: "El sistema debe procesar 100 solicitudes simultáneas con un tiempo de respuesta inferior a 2 segundos".

Elicitación de requerimientos:  
 Es el proceso de recopilar información para definir claramente los requisitos del sistema. Técnicas comunes incluyen:

Entrevistas con stakeholders.

Talleres de trabajo colaborativo.

Encuestas o cuestionarios.

Observación directa.

Prototipos o maquetas.

Análisis de documentación existente.

¿Qué es un caso de uso y cómo se utiliza en el análisis funcional?  
 Es una descripción detallada de cómo los usuarios interactúan con el sistema para lograr un objetivo específico.  
 Ejemplo:

Actor: Usuario.

Caso de uso: Registrarse en la plataforma.

Flujo principal: El usuario introduce su correo, elige una contraseña, y confirma. El sistema valida y crea la cuenta.

Importancia de la priorización de requerimientos:  
 Permite enfocarse en los aspectos más valiosos y urgentes del sistema. Factores a considerar incluyen:

Impacto en los objetivos del negocio.

Valor para el usuario.

Complejidad técnica.

Riesgos asociados.

Tema 2: Metodologías Ágiles

¿Qué son las metodologías ágiles y en qué se diferencian de las tradicionales?  
 Son enfoques iterativos y colaborativos para el desarrollo de software, enfocadas en adaptarse rápidamente a los cambios. Se diferencian de las tradicionales, como el modelo en cascada, en que priorizan la entrega continua de valor y la retroalimentación constante.

Cuatro valores fundamentales del Manifiesto Ágil:

Individuos e interacciones sobre procesos y herramientas.

Software funcionando sobre documentación extensiva.

Colaboración con el cliente sobre negociación de contratos.

Responder al cambio sobre seguir un plan.

Scrum y sus elementos principales:

Sprint: Ciclos de trabajo cortos (1-4 semanas) para entregar incrementos funcionales.

Product Backlog: Lista priorizada de tareas y funcionalidades.

Scrum Master: Facilita el proceso y elimina obstáculos para el equipo.

¿Qué es un "sprint" y su propósito?  
 Es un intervalo de tiempo fijo en Scrum donde el equipo desarrolla un conjunto de funcionalidades incrementales. Su propósito es entregar valor de manera continua y recibir retroalimentación.

¿Qué es una "historia de usuario"?  
 Una descripción breve y clara de una funcionalidad desde la perspectiva del usuario final.  
 Ejemplo: "Como usuario, quiero buscar productos por nombre para encontrarlos más rápido."

Tema 3: Ciclo de Vida del Software

¿Qué es el ciclo de vida del software?  
 Es el conjunto de etapas necesarias para desarrollar, desplegar y mantener un sistema. Fases principales:

Análisis de requisitos.

Diseño.

Desarrollo.

Pruebas.

Implementación.

Mantenimiento.

Diferencia entre modelo en cascada y modelo iterativo:

Modelo en cascada: Fases secuenciales, cada etapa debe completarse antes de pasar a la siguiente.

Modelo iterativo: Desarrollo en ciclos repetitivos, permite retroalimentación y ajustes constantes.

Fase de pruebas:  
 Es la etapa donde se verifica que el software cumpla con los requisitos definidos. Es crucial porque asegura la calidad y evita errores en producción.

¿Qué es el mantenimiento del software?  
 Es la etapa donde se corrigen errores, mejoran funcionalidades y se adaptan cambios tras la implementación. Es importante porque garantiza la vigencia y eficacia del sistema.

Fase de análisis de requisitos:  
 Es el proceso de recopilar, documentar y validar las necesidades de los stakeholders. Es fundamental porque define la base para el diseño y desarrollo del sistema, asegurando que cumpla con los objetivos.