

PREGUNTAS VERDADERO/FALSO PROCESO SOFTWARE

1. Las pruebas estáticas se usan para que aprendan los programadores jóvenes. F
2. El modelo de análisis, forma gráfica, sirve para saber que quiere el analista. F
3. Un ciclo de vida indica el orden de los procesos software. V
4. La prueba negra se usa para validar el funcionamiento del programa. V
5. El tiempo se mide en recurso por hombre. F
6. Fiabilidad capacidad de ofrecer los mismos resultados bajo las mismas condiciones. V
7. Un hito no consume tiempo. V
8. En el modelo V las ramas descendentes se usan para realizar pruebas. F
9. El programador realiza toda la documentación del proyecto. F
10. Requisitos funcionales son las cualidades del software. F
11. Tablas interfaz. JA
12. La calidad es un método de prevención. V
13. Toda la configuración de un programa de alto nivel es externa. F
14. Un buen diseño debe perseguir la mínima cohesión y el máximo acoplamiento. F
15. La forma que tenga el DFD nos va a indicar que tipo de estrategia debemos de usar para obtener el diseño correspondiente. V
16. La comunicación directa entre entidades extremas nunca se reflejará en un DFD. V
17. El DFD debe estar referido a aspectos físicos del sistema, ya que el EP puede Describirlos. F
18. En las pruebas de caja negra se usan conductores y resguardos. JA
19. Las revisiones de código solo se pueden realizar sobre programas nuevos. F
20. En el aseguramiento de calidad se actúa para encontrar no conformidades. V
21. La fase de definición nos indica como construir el sistema. F
22. En el modelo de ciclo de vida en espiral, cada ciclo en la espiral representa una fase del proceso de análisis. V
23. Un requisito es una característica que el sistema debe tener, pero no es una restricción que el sistema debe satisfacer para ser aceptada por el cliente. F
24. Los modelos de análisis son elementos contra los que validar el software. V
25. Los requisitos no funcionales hacen que el producto sea atractivo, útil, rápido, fiable o Seguro. V

26. Es correcta la afirmación "Prueba, luego codifica". F
27. El control de calidad pone el foco en los procesos. F
28. El software se puede definir como el conjunto de dos componentes: instrucciones y estructuras de datos. F
29. El diagrama de estructuras describe como cada módulo del sistema realiza la función que se le encomendó. F
30. El DFD no maneja en ningún momento el concepto de caja negra. F
31. Siempre que se realice la prueba de unidad (caja blanca) son necesarios conductores y resguardos. JA
32. Las revisiones estáticas son un buen medio para la evaluación de los recursos. JA
33. El objetivo de la prueba es prevenir errores antes de que ocurra. V
34. Comúnmente se usa la generalización para actores y no para casos de uso. V
35. Los modelos de análisis permiten verificar que el analista ha comprendido el problema o entorno a abordar. V
36. La siguiente definición puede formar parte de un DD: $a - @b - @c + \{ @d \} + \{ [e | f | g] \}$ 10. F
37. Toda la configuración de un programa la realiza el programador. F
38. En el modelo de ciclo de vida es espiral: se pone especial énfasis en el análisis de los riesgos para reducir su impacto. V
39. En un DFD una entidad externa no tendrá acceso directo a los datos de un almacén. V
40. Un requisito software es una característica que el sistema debe tener, pero no es una restricción que el sistema deba satisfacer para ser aceptada por el cliente. F
41. Una prueba no tiene límites. V
42. Definición de hamaca
43. Definición de requisito
44. El ciclo de vida marca lo necesario para pasar al siguiente proceso. V
45. El ciclo de vida de un sistema es el mismo que el de un programa. F
46. Un nodo de procesamiento tiene dos salidas. F
47. La documentación de un programa solo la hace un programador. F
48. El DFD nos dice como será el tipo de diseño. V
49. Para describir la interfaz es necesaria la tabla interfaz. F
50. Definición calendarización (¿Qué es lo que marca?). JA

51. Si realizamos una prueba y no encuentra errores decirnos que es una prueba

Satisfactoria. F

52. La prueba está orientada a prevenir errores. V

53. Un proceso es un marco de trabajo. V

54. El aseguramiento de la calidad se hace sobre un proceso. V

PREGUNTAS CORTAS

1. Cita las 3 restricciones de un proyecto

fecha de entrega requerida, alcance, presupuesto

2. Clasificar los tipos de prueba en función de objetivos

de unidad , de integración, de validación y de sistema

3. Enumera los tipos de mantenimiento

Correctivo: un programa no realiza correctamente la aplicación para la que ha sido diseñado, y, por tanto, debe ser modificado.

Perfectivo: modificaciones a los programas para conseguir mayor adecuación a los requisitos, mayor eficiencia, o simplemente recoger nuevas funcionalidades no expresadas en la fase de definición del sistema

Adaptativo: Adaptar los programas para acomodarlos a los cambios de su entorno externo

Preventivo: El software se deteriora con los cambios, y este tipo de mantenimiento hace cambios en los programas para que se puedan corregir, adaptar y mejorar más fácilmente

4. Enumera los elementos de un DFD

Entidad externa, flujo, proceso y almacén.

5. Enumera los tipos de validación dinámica

6. Actividades de la ingeniería de requisitos

Obtención (Análisis del problema), Especificación (documentar los requisitos), Validación (que sean consistentes y estén completos)

7. Metodologías ágiles

SCRUM

8. Metodologías tradicionales

MetricaV3

9. Problemas del desarrollo software

- Proyectos fuera de plazo y de presupuesto
- Excesiva dependencia de los desarrolladores
- Falta de control del desarrollo del proyecto
- Escasa integración de las diferentes fases del desarrollo
- Escaso control de calidad del producto
- Escasa documentación actualizada de los proyectos
- No utilizar una metodología formal

10. Tipos de relaciones en los diagramas de casos de uso

Asociación, extiende, generalización, inclusión

11. Tipos de requisitos

Requisitos de Negocio: Requisitos a más alto nivel. Representan los objetivos, base de negocio, estrategia, alcance...

Requisitos de Usuario: Los requisitos de cliente o usuario, deben describir los requisitos del producto de tal forma que sean comprensibles por los propios usuarios del sistema sin conocimiento técnico previo

Requisitos del Sistema/Software: Utilizados por los ingenieros de software como punto de partida para el diseño del producto

12. Capas de la tecnología multicapa

Herramientas, métodos, proceso

13. Principios de la ingeniería software

Abstracción, encapsulamiento, completitud, uniformidad, modularidad, localización, validación y verificación.

14. Fases del desarrollo software

Definición, desarrollo, mantenimiento

15. Características ciclos de vida orientados a objetos