



Documento anónimo

RespuestasFIS.pdf

+65 Preguntas Tipo Test Con Solución



2º Fundamentos de la Ingeniería del Software



Grado en Ingeniería Informática



Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Universidad de Granada

Maths
informática

**Tan lejos como nunca, tan cerca
como siempre**

CURSOS INTENSIVOS en JUNIO para todas las asignaturas de
Ingeniería Informática

615 29 80 22 91 399 45 49 C/Andrés Mellado, 88 duplicado

academia.maths www.mathsinformatica.com academia@mathsinformatica.com



- 1.- Los modelos del análisis pueden contener tantas inconsistencias como consideremos oportunas, puesto que no son la solución del problema. F
- 2.- Uno de los objetivos del análisis es conseguir los requisitos del software a partir de los requisitos de usuario mediante un proceso de refinamiento. V
- 3.- Un Diagrama de Secuencia del Sistema se puede corresponder con un caso de uso, con un diagrama de casos de uso o con todo el sistema. V
- 4.- El nombre que le demos al sistema en el DSS se va a corresponder con el nombre de una clase que va a formar parte de nuestra solución. V
- 5.- El Contrato de una operación debe indicar qué hace una operación sin decir cómo lo hace. V
- 6.- Los modelos de AER son: Modelo conceptual, diagramas de casos de uso y los contratos de las operaciones principales. F
- 7.- El modelo conceptual debe representar cualquier tipo de relación que se dé entre los conceptos que forman parte de él. F
- 8.- Un concepto debe incluir los atributos que indique las asociaciones que tienen otros conceptos. F
- 9.- En un contrato si está relleno el apartado de las excepciones, el apartado de las precondiciones debe estar vacío. F
- 10.- Lo siguiente es una poscondición correcta: "se creó una lista en la que se incluye el nombre del cliente, dirección y teléfono, que se proporciona como salida de la operación". F
- 11.- A la hora de elaborar el diagrama de comunicación de una operación son esenciales los siguientes apartados del contrato correspondiente: excepciones, precondiciones y poscondiciones. F
- 12.- Con la abstracción de datos se abstraen sobre el funcionamiento para conseguir una estructura modular basada en procedimientos. F
- 13.- EL acoplamiento es un indicador de la dependencia entre módulos, cuanto más alto sea este valor mejor será el diseño. F
- 14.- El análisis de la productividad permite realizar una buena gestión de proyectos. V
- 15.- El diagrama de clases de diseño se deduce de los diagramas de comunicación. Elaboran los diagramas de comunicación y después el diagrama de clases del diseño. F
- 16.- El diagrama de secuencia del sistema puede contener tantos objetos como sean necesarios para llevar a cabo una operación del sistema. F
- 17.- EL diseño es el proceso de refinamiento, en el que partiendo de modelos del análisis vamos añadiendo información hasta completar el diseño. V
- 18.- El mayor esfuerzo durante el proceso de producción del software se realiza en la etapa de desarrollo. F
- 19.- El mayor esfuerzo realizado durante el mantenimiento de un software es para adaptar el software o nuevos requisitos. F
- 20.- El modelado de casos de uso solo puede ser usado en la etapa de detección de requisitos. F
- 21.- El modelo conceptual no debe incluir los nombres de rol de las asociaciones. F
- 22.- El modelo conceptual no puede contener las navegabilidades de las asociaciones. V
- 23.- El modelo conceptual o modelo de dominio es básico para especificar las



postcondiciones de un contrato. V

24.- El modelo de casos de uso permite determinar con facilidad los requisitos no funcionales del sistema. F

25.- El modelo de casos de uso se usa exclusivamente para la obtención de requisitos. F

26.- El número de módulos de un sistema software debe ser cuantos más mejor, pues así garantizamos la independencia modular de cada uno de ellos. F

27.- El número de operaciones principales de un sistema es el mismo que el número de casos de usos que tengamos. F

28.- El proceso unificado es un modelo de proceso dirigido por casos de uso. V

29.- El resultado del diseño de la arquitectura del software es un conjunto de subsistemas y las relaciones entre ellos. V

30.- En el diagrama de clases del diseño pueden aparecer nuevas clases, es decir, que no estén en el modelo conceptual. V

31.- En el modelo conceptual hay que definir los atributos y los métodos de todas las clases. V

32.- En los diagramas de clases de diseño no se deben representar las relaciones de dependencia entre clases, solo se deben representar las de asociación y de generalización. F

33.- En los diagramas de clases de diseño pueden aparecer relaciones de dependencia. V

34.- En un diagrama de secuencia del sistema pueden aparecer tantos objetos como necesitemos para modelar la interacción entre ellos y con los actores. F

35.- Es posible que en un caso de uso no tenga que intervenir el sistema software a modelar. F

36.- La arquitectura cliente-servidor favorece la escalabilidad de los sistemas software, porque permite la reconfiguración añadiendo clientes y servidores extra. V

37.- La forma más directa de identificar casos de uso es identificando los objetivos y necesidades de los actores del sistema. V

38.- La navegabilidad de las asociaciones en el diagrama de clases del diseño se obtiene teniendo en cuenta la dirección en los envíos de mensaje en los diagramas de comunicación. V

39.- La primera tarea del diseño es encontrar el diseño de la arquitectura del sistema. V

40.- Las relaciones entre actores y casos de uso son la asociación y la dependencia. F

41.- Las relaciones entre los casos de uso pueden ser asociación, generalización y dependencia. F

42.- Las relaciones que se dan entre casos de uso es la dependencia y la generalización. V

43.- Las tareas principales de la ingeniería de requisitos son detección, análisis, especificación, revisión y reacción de requisitos. V

44.- Las vías de comunicación o enlaces entre objetos en un diagrama de colaboración son bidireccionales. V

45.- Lo siguiente es un recurso funcional "las reservas de préstamos de libros caducan a los 10 días a partir del momento que el libro esté a disposición del



usuario". V

46.- Lo siguiente es un requisito NO funcional de facilidad de uso "el entorno debe avisar al usuario mediante email tras días antes de que finalice el plazo del préstamo". F

47.- Los siguiente es una poscondición correcta "se creó una lista en la que se incluye el nombre del cliente, dirección y teléfono. Que se proporciona como salida de la operación". F

48.- Los actores representan roles que son interpretados por personas, dispositivos, otros sistemas... cuando el sistema está en uso. V

49.- Los actores tienen que ser necesariamente los identificados como usuarios del sistema. F

50.- Los diagramas de actividad de UML es una herramienta muy adecuada para el diseño del flujo de control. V

51.- Los diagramas de interacción y los diagramas de actividad UML son herramientas de diseño que permiten representar lo mismo, son equivalentes. F

52.- Los modelos de AER son: modelo conceptual, diagramas de casos de uso y los contratos de las operaciones principales. F

53.- Los prototipos siempre se transforman hasta convertirse en el programa que se entrega al cliente. F

54.- Los requisitos no funcionales determinan los objetivos del diseño. F

55.- Los requisitos no funcionales suponen limitaciones para el diseño de un sistema software. V

56.- Los tipos de requisitos son funcionales, no funcionales y FURPS+. F

57.- Para elaborar el modelo de análisis es fundamental el modelo de casos de uso. V

58.- Para incorporar generalizaciones es necesario encontrar clases conceptuales con elementos comunes. V

59.- Si una función del sistema no cambia nada de lo especificado en el modelo conceptual su contrato no tendrá poscondiciones. V

60.- Un caso de uso esencial describe que hace el sistema como respuesta a una petición de algún actor, pero no como lo hace. V

61.- Un caso de uso produce algo de valor para un actor. V

62.- Un caso de uso puede ser iniciado por un actor o por un usuario. F

63.- Un concepto no debe incluir los atributos de otros conceptos que indiquen las relaciones entre ellos. V

64.- Un modelo de casos de uso lo componen los diagramas de casos de uso y la especificación de actores y casos de uso. V

65.- Un modelo de casos de uso se centra en las necesidades que el usuario espera lograr al utilizar el sistema. V

66.- Un nivel de acoplamiento alto y de cohesión bajo en un módulo garantiza un diseño de calidad. F

67.- Una asociación es una conexión significativa y relevante entre conceptos. V

68.- Una mala solución para remediar el retraso en la entrega de un proyecto software es la llamada "horda mongoliana". V