

[FIS] Examen

1. Todos los sustantivos que se identifiquen en los casos de uso se representan como conceptos en el diagrama conceptual. **F**

Los sustantivos serán los candidatos a conceptos del diagrama conceptual, pero no todos serán conceptos.

2. El diagrama de componentes especifica el hardware físico sobre el que se ejecutará el sistema software. **F**

No, el que se encarga de esto es el diagrama de despliegue. El diagrama de componentes representa una estructuración concreta del sistema a partir de los componentes software y su interrelación.

3. Unos de los objetivos de la fase de inicio del proceso unificado es el estudio de viabilidad del sistema a desarrollar. **V**

El proceso unificado es un modelo híbrido que apoya la creación de prototipos y la entrega incremental. Identifica cuatro fases en el proceso de desarrollo software y una de ellas es la concepción, donde el objetivo es establecer un caso empresarial para el sistema.

4. Durante la etapa de definición hay que conseguir encontrar la solución software al sistema analizado. **F**

Encontrar la solución software al sistema analizado se corresponde a la etapa de construcción.

5. Las clases del diagrama de clases del diseño toman sus atributos de los diagramas de comunicación. **F**

Las clases identificadas tomarán sus atributos del modelo conceptual y de los diagramas de interacción.

6. Todos los enlaces estereotipados con <<L>>, <<P>> o <<G>> estarán en el diagrama de clases del diseño como una asociación. **F**

Todos los enlaces estereotipados con <<L>>, <<P>> o <<G>> estarán en el diagrama de clases del diseño como una dependencia.

7. El usuario es una pieza importante en el proceso de validación de las especificaciones del software. **V**

En la validación revisamos todo lo que se ha obtenido junto con la prueba del código. Por tanto, debemos asegurarnos que definen el sistema que el cliente y los usuarios desean.

8. El uso de mecanismos de abstracción en el diseño permiten obtener la modularidad adecuada de un sistema software. **F**

Mecanismo que permite determinar qué es relevante y qué no lo es en un nivel de detalle determinado, ayudando a obtener la modularidad adecuada para ese nivel de detalle (no es lo mismo ayudar que permitir).

9. Uno de los problemas más importantes en el proceso de desarrollo del software es el incumplimiento de la planificación. **V**

Problemas en el proceso de desarrollo del software: comunicación entre personas, incumplimiento de la planificación e incorporar cambios en etapas avanzadas del proceso.

10. El modelo de prototipos es un buen método para validar los requisitos de los usuarios en cualquier proyecto de desarrollo de software. **F**

No es el mejor enfoque para algunos tipos de sistemas.

11. En la arquitectura multicapa las capas deben estar lo más acopladas posible. **F**

Arquitectura en capas: Organiza el sistema en capas con funcionalidad relacionada con cada capa. Una capa da servicios a la capa de encima y las capas de nivel inferior representan servicios núcleo que es probable que se utilicen a lo largo de todo el sistema.

Principios de diseño: las capas deben estar desacopladas, ...

12. Los requisitos no funcionales definen los criterios de calidad del sistema software. **V**

Los requisitos no funcionales especifican aspectos técnicos que debe incluir un sistema. Pueden clasificarse en: restricciones y calidades.

13. Las asociaciones de navegación se obtienen a partir de las asociaciones del modelo conceptual. **F**

14. Los requisitos no funcionales no tienen ninguna relación con los funcionales. **F**

Los requisitos funcionales especifican las funciones que un sistema, o componente de un sistema, debe ser capaz de llevar a cabo. En cambio, los requisitos no funcionales especifican aspectos técnicos que deben incluir un sistema.

15. En el diagrama de clases del diseño pueden aparecer clases que no estaban en el diagrama de conceptos construido en el modelo de análisis. **V**

En el diagrama de clases, los atributos que se usen deben aparecer en el modelo conceptual y los diagramas de interacción pero podrán aparecer clases no estipuladas en el modelo conceptual si se requieren para la solución del problema.

16. Una de las funciones de la relación de inclusión en los casos de uso es descomponer un caso de uso complejo y largo en varios, para facilitar su comprensión. **V**

Inserción de comportamiento adicional dentro del caso de uso base que explícitamente hace referencia al caso de uso de inclusión.

17. La detección de conflictos entre los requisitos es una de las principales actividades del análisis de requisitos. **V**

Análisis de requisitos: Se trata fundamentalmente de detectar y resolver conflictos entre requisitos, delimitar el software y establecer con qué elementos externos interacciona y elaborar los requisitos del sistema para obtener, a partir de ellos, los requisitos del software a desarrollar

18. Las bases principales para obtener los diagramas de comunicación son los contratos y el modelo conceptual. **V**

19. Un caso de uso sólo puede tener un actor principal que coincide con el que inicia el caso de uso. **F**

Los actores principales además de interactuar con el caso de uso son los que lo activan. Pueden ser varios.

20. El uso de métodos de desarrollo ágiles rompen con la filosofía de equipos de trabajo organizados de forma jerárquica. **V**

Métodos ágiles: Conjunto de métodos que enfatizan el enfoque iterativo, la adaptabilidad del proceso y la colaboración, reduciendo al mínimo la documentación y los procedimientos.

Características: No hay una estructura organizativa rígida en el equipo de desarrollo, ...

21. Una de las ventajas al incluir las relaciones entre los casos de uso es que se reduce el texto generado en la descripción de los casos de uso. **V**

Usar las relaciones entre casos de uso cuando simplifiquen el modelo.

22. Los diagramas de actividad se usan como complemento a la descripción de un caso de uso complejo. **V**

Los diagramas de actividad se pueden usar para representar los flujos de un caso de uso de forma gráfica.

23. El uso del patrón controlador en la elaboración del modelo de diseño se hace para reducir el nivel de acoplamiento entre los elementos de la interfaz de usuario y los que modelan la solución. **V**

24. Los proyectos software reales raramente se adaptan a un modelo de ciclo de vida clásico o en cascada. **V**

Esto es así ya que es inflexible en la división del proyecto en distintas etapas.

25. El análisis de requisitos permite descubrir los conflictos existentes entre los requisitos. **V**

Análisis de requisitos: Se trata fundamentalmente de detectar y resolver conflictos entre requisitos, delimitar el software y establecer con qué elementos externos interacciona y elaborar los requisitos del sistema para obtener, a partir de ellos, los requisitos del software a desarrollar

26. Durante el análisis no se estudia la solución que se va a proponer al problema planteado, eso se deja a la fase de diseño. **V ???**

Fase de diseño: Búsqueda de una solución. Descripción de los componentes, sus relaciones y funciones.
Fase de análisis: Análisis del problema a resolver. Documento en el que se indica la funcionalidad del software y las restricciones sobre su operación.

27. Con el análisis orientado a objetos sólo se modelan las propiedades estáticas del ámbito del problema. **F**

Los métodos de análisis orientado a objetos se centran en obtener dos tipos de modelos: estáticos o de estructura y dinámicos o de comportamiento.

28. Los casos de uso "esenciales" son los procedimientos comunes más importantes del sistema. **F**

Esto ocurren en los casos de uso primarios.

29. No se deben usar atributos de un concepto como clave de acceso desde otro concepto. **V**

No usar atributos de un concepto como clave de acceso desde otro concepto.

30. El modelo conceptual se representa usando un diagrama de clases que contiene las clases con sus atributos, métodos y asociaciones. **F**

El modelo conceptual se utiliza como guía (no contiene atributos...)

31. En un diagrama de secuencia del sistema pueden aparecer tantos objetos como se necesiten para modelar la interacción entre ellos. **F**

Los objetos que aparecen son los modelados en los pasos anteriores, no pueden existir nuevos.

32. Un caso de uso puede generar más de una operación en el diagrama de secuencia del sistema. **V**

Esto lo hemos visto haciendo las prácticas.

33. Los patrones de diseño para la asignación de responsabilidades a objetos ayudan a obtener el diagrama de clases de diseño. **F**

Los patrones de diseño para la asignación de responsabilidades a objetos ayudan al modelo de interacción de objetos.

34. Una asociación es una conexión significativa y relevante entre conceptos. **V**

Teoría: Una asociación es una conexión significativa y relevante entre conceptos.

35. El modelo estructural del análisis está representado por el/los diagramas de secuencia del sistema. **F**

El modelo de análisis utiliza como guía el modelo conceptual para su estructura y para su comportamiento con el diagrama de secuencias y los contratos.

36. Antes de definir una subclase en un modelo conceptual se debe comprobar que cumple las reglas del 100% y del "es-un". **V**

Para definir una subclase en un modelo conceptual se deben cumplir dichas reglas.

37. El diseño es una tarea clave para la calidad del producto software. **V**

Es una de las etapas.

38. Un cambio de estado que se describe en las postcondiciones de un contrato es la creación de un atributo. **F**

Las poscondiciones deben describir los cambios de estado de un sistema no sus acciones, estos son: creación y destrucción de objetos, creación y destrucción de enlaces y modificación de atributos.

39. Cuando se construye un modelo conceptual es mejor añadir el mayor número posible de asociaciones entre conceptos. **F**

Tienes que colocar las que necesitas.

40. Un participante en un diagrama de secuencia puede ser un objeto individual o un multiobjeto. **V**

????? es así por ley, pa que sepan

41. La diferencia entre una precondition y una excepción es que la precondition no tiene que comprobarse en la operación que se está definiendo. **V**

La diferencia de la precondition y la excepción es que la precondition se comprueba antes de iniciar la operación y la excepción una vez iniciada.

42. En el diagrama de clases de diseño, la multiplicidad se obtiene de la existencia o no de multiobjetos en los diagramas de comunicación. **V**

La navegabilidad la da la dirección del envío de mensaje y la multiplicidad la existencia de multiobjetos.

43. La Etnografía es una técnica de obtención de requisitos que consiste en preguntar a los trabajadores de un negocio sobre la forma en que realizan sus tareas. **F**

Las técnicas etnográficas son técnicas de observación que se usan para entender los procesos operacionales y ayudar a derivar requisitos sociales y de organización, de apoyo para dichos procesos.

44. El incumplimiento de la planificación lleva de forma inmediata al aumento de personal en el equipo de desarrollo. **F**

No debemos recurrir a la "horda mongoliana". Si no sabes lo que es, haber estudiado.

45. Una característica de los métodos ágiles es las entregas frecuentes. **V**

La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporten un valor.

46. Un diagrama de secuencia del sistema es un diagrama de secuencia de UML en el que se muestran los eventos generados por los actores. **V**

Diagrama de secuencia de UML en el que se muestran cómo los eventos generados por los actores provocan la ejecución de una operación por el sistema, siendo visto éste como una caja negra.

47. Los requisitos de un proyecto software pueden cambiar continuamente, pero esto no es un problema ya que los sistemas software son flexibles (se adaptan a los cambios). **F**

Incorporar cambios en etapas avanzadas del proceso es un problema.

48. Para obtener un buen diseño, cada módulo debe presentar un bajo nivel de cohesión. **F**

La independencia se evalúa con dos criterios: cohesión (alta cohesión es lo ideal) y acoplamiento (bajo acoplamiento es lo ideal).

49. La arquitectura de un sistema software facilita la comprensión de la estructura global del sistema. **V**

Importancia de la arquitectura: facilita la comprensión de la estructura global del sistema, permite trabajar en los componentes de forma independiente, facilita las posibles extensiones del sistema y facilita la reutilización de los distintos componentes.

50. Uno de los pasos a realizar en la elaboración del modelo de interacción de objetos es la incorporación de las asociaciones entre las clases de objetos. **F**

Pasos a seguir para el diseño de los casos de uso:

- A. Elaborar los diagramas de interacción (ninguno de los pasos coincide con este).
- B. Inicialización del sistema.
- C. Establecer relaciones entre el modelo y la Interfaz de Usuario.

51. Los modelos del análisis pueden contener tantas inconsistencias como consideremos oportunas, puesto que no son la solución del problema. **F**

Se busca que no hayan inconsistencias.

52. - 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**

Objetivo principal del análisis de requisitos: refinar, estructurar y describir los requisitos para una comprensión más precisa y fácil de mantener que ayude a estructurar el sistema completo (modelos del análisis).

53. 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**

54. 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**

Hecho en prácticas.

55. El contrato de una operación debe indicar qué hace una operación sin decir cómo lo hace. **V**

Documento que describe lo que una operación se propone lograr, sin decir cómo se conseguirá.

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

Diagrama de clases de UML (modelo conceptual), diagrama de secuencia de UML y contratos de C. Larman.

57. El modelo conceptual debe representar cualquier tipo de relación que se dé entre los conceptos que forman parte de él. **F**

Se añaden las asociaciones, que son las relaciones más significativas.

58. Un concepto debe incluir los atributos que indique las asociaciones que tienen otros conceptos. **F**

59. En un contrato si está relleno el apartado de las excepciones, el apartado de las precondiciones debe estar vacío. **F**

60. Lo siguiente es una postcondición 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**

La oración debe estar totalmente en pasado.

61. A la hora de elaborar el diagrama de comunicación de una operación son esenciales los siguientes apartados del contrato correspondiente: excepciones, precondiciones y postcondiciones. **F**

Todo lo especificado en el contrato, especialmente las poscondiciones, excepciones y salidas, tienen que satisfacerse en el correspondiente diagrama de comunicación.

62. Con la abstracción de datos se abstraen sobre el funcionamiento para conseguir una estructura modular basada en procedimientos. **F**

63. EL acoplamiento es un indicador de la dependencia entre módulos, cuanto más alto sea este valor mejor será el diseño. **F**

Cuanto menor.

64. El análisis de la productividad permite realizar una buena gestión de proyectos. **V**

Esto es más verídico que yo follando a la madre de Josemi.

65. 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**

Se deduce del modelo conceptual y del modelo de interacción de objetos.

66. El diagrama de secuencia del sistema puede contener tantos objetos como sean necesarios para llevar a cabo una operación del sistema. **F**

67. EL diseño es el proceso de refinamiento, en el que partiendo de modelos del análisis vamos añadiendo información hasta completar el diseño. **V**

Refinar, estructurar y describir los requisitos para una comprensión más precisa y fácil de mantener que ayude a estructurar el sistema completo (modelos del análisis).

68. El mayor esfuerzo durante el proceso de producción del software se realiza en la etapa de desarrollo. **F**

Es la etapa de mantenimiento.

69. El mayor esfuerzo realizado durante el mantenimiento de un software es para adaptar el software o nuevos requisitos. **F**

Es cambios en los requisitos del usuario.

70. El modelado de casos de uso solo puede ser usado en la etapa de detección de requisitos. **F**

71. El modelo conceptual no debe incluir los nombres de rol de las asociaciones. **F**

Si tiene dichos nombres.

72. El modelo conceptual no puede contener las navegabilidades de las asociaciones. **V**

La navegabilidad se incorpora en los daigramas de clase del diseño, que viene después.

73. El modelo conceptual o modelo de dominio es básico para especificar las postcondiciones de un contrato. **V**

Lo hemos hecho en las prácticas.

- 74.- El modelo de casos de uso permite determinar con facilidad los requisitos no funcionales del sistema. **F**
- 75.- El modelo de casos de uso se usa exclusivamente para la obtención de requisitos. **F**
- 76.- 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**
- 77.- El número de operaciones principales de un sistema es el mismo que el número de casos de usos que tengamos. **F**
- 78.- El proceso unificado es un modelo de proceso dirigido por casos de uso. **V**
- 79.- El resultado del diseño de la arquitectura del software es un conjunto de subsistemas y las relaciones entre ellos. **V**
- 80.- En el diagrama de clases del diseño pueden aparecer nuevas clases, es decir, que no estén en el modelo conceptual. **V**
- 81.- En el modelo conceptual hay que definir los atributos y los métodos de todas las clases. **V**
- 82.- 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**
- 83.- En los diagramas de clases de diseño pueden aparecer relaciones de dependencia. **V**
- 84.- En un diagrama de secuencia del sistema pueden aparecer tantos objetos como necesitemos para modelar la interacción entre ellos y con los actores. **F**
- 85.- Es posible que en un caso de uso no tenga que intervenir el sistema software a modelar. **F**
- 86.- La arquitectura cliente-servidor favorece la escalabilidad de los sistemas software, porque permite la reconfiguración añadiendo clientes y servidores extra. **V**
- 87.- La forma más directa de identificar casos de uso es identificando los objetivos y necesidades de los actores del sistema. **V**

- 88.- 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**
- 89.- La primera tarea del diseño es encontrar el diseño de la arquitectura del sistema. **V**
- 90.- Las relaciones entre actores y casos de uso son la asociación y la dependencia. **F**
- 91.- Las relaciones entre los casos de uso pueden ser asociación, generalización y dependencia. **F**
- 92.- Las relaciones que se dan entre casos de uso es la dependencia y la generalización. **V**
- 93.- Las tareas principales de la ingeniería de requisitos son detección, análisis, especificación, revisión y reacción de requisitos. **V**
- 94.- Las vías de comunicación o enlaces entre objetos en un diagrama de colaboración son bidireccionales. **V**
- 95.- 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**
- 96.- 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**
- 97.- 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**
- 98.- Los actores representan roles que son interpretados por personas, dispositivos, otros sistemas... cuando el sistema está en uso. **V**
- 99.- Los actores tienen que ser necesariamente los identificados como usuarios del Sistema. **F**
- 100.- Los diagramas de actividad de UML es una herramienta muy adecuada para el diseño del flujo de control. **V**
- 101.- Los diagramas de interacción y los diagramas de actividad UML son herramientas de diseño que permiten representar lo mismo, son equivalentes. **F**

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Otras preguntas de informática

1. Análisis de requisitos es sinónimo de ingeniería de requisitos. **F**
2. El análisis y especificación de requisitos es cuando el sistema se hace más fácil de entender para los desarrolladores. **V**
3. Una de las partes importantes de las que se encarga el análisis de requisitos es en facilitar el mantenimiento del sistema. **V**
4. El modelo de análisis se estructura en casos de uso. **F**
5. En el modelo de casos de uso, era correcto usar numerosas redundancias. Sin embargo, hay que intentar evitarlas en el modelo de análisis. **V**
6. El análisis orientado a objetos tiene dos enfoques, estático (abstrae solamente las cualidades de los objetos) o dinámico (abstrae solamente el comportamiento de los objetos). **V**
7. Con el análisis de requisitos pasamos del que queremos del sistema a como queremos implementarlo. **V**
8. Las herramientas para representar los modelos de análisis son el diagrama de clases de UML, diagramas de casos de uso y contratos de C. Larman. **F**
9. Modelo estático es sinónimo de diagrama de conceptos y modelo conceptual. **V**
10. El modelo de casos de uso se usa como base para obtener la información necesaria para el diagrama de clases. **V**
11. Para conseguir el modelo estático, tenemos que buscar los conceptos, las asociaciones (entre conceptos), generalizaciones (de conceptos) y atributos (de conceptos). **V**
12. Los adjetivos o frases adjetivales tienden a ser candidatos a conceptos. **F**
13. Buscar relaciones de generalización consiste en buscar posibles clases que pudieran ser abstractas. **F**
14. Para crear una subclase, esta debe de tener un atributo o asociación diferente que la superclase. **T**
15. Un diagrama de secuencia del sistema es un diagrama UML en el que se muestra la secuencia desencadenada en el sistema tras la intervención de un actor. **V**
16. Los contratos explican en lenguaje natural la implementación que debe tener una operación. **F**
17. Las postcondiciones en los contratos se deben escribir en futuro. Por ejemplo "se incrementará el precio del producto". **F**