

Preguntas ISOF

1. Introducción	4
1. ¿Qué es la ingeniería del software?.....	4
2. ¿Qué son los procesos de software? Enumera qué procesos genéricos se abordan en un proyecto de desarrollo de software, indicando brevemente en qué consiste cada uno de ellos.	4
2. Metodologías de desarrollo de software	4
1. ¿Cuáles son los procesos principales de Métrica 3? Descríbelos brevemente.....	4
2. Realiza un diagrama GANTT que represente los procesos principales de Métrica 3 en un proyecto de desarrollo de software y descríbelos brevemente.	5
3. ¿Cuál es el objetivo del EVS?	5
4. ¿En qué orden se deben realizar las actividades del proceso EVS?	6
5. ¿Qué se hace y qué se obtiene en la actividad EVS 1: establecimiento del alcance del sistema?.....	6
6. ¿Qué debe estudiarse en el EVS respecto a la situación actual?	6
7. ¿Qué debe hacerse en el EVS respecto al análisis y valoración de la alternativa de construcción?.....	6
8. ¿Cuál es el objetivo del ASI?	7
9. Actividades del proceso ASI.....	7
10. Realiza un diagrama GANTT que represente las actividades del proceso ASI y descríbelas brevemente	8
11. Cita y describe brevemente los modelos utilizados en el proceso ASI.....	8
12. ¿Qué productos se obtienen en la actividad del ASI denominada “Definición de interfaces de usuario”?	9
13. ¿Cuáles son los objetivos de la actividad del ASI denominada Análisis de Consistencia y Especificación de Requisitos?	9
14. ¿Qué actividades del proceso ASI de la metodología Métrica 3 no se abordan en las prácticas de la asignatura?	9
15. Resume los objetivos y productos más importantes del proceso de Diseño del Sistema de Información de la metodología Métrica 3	9
16. ¿En qué proceso de la metodología Métrica se codifica el software? Menciona los productos fundamentales de ese proceso, algunas de sus actividades y con qué proceso continúa la metodología	10
17. ¿Cuáles son los objetivos principales del proceso de Implantación y Aceptación del Sistema en la metodología Métrica? ..	10
18. ¿Qué es un ciclo de vida incremental? ¿Qué diferencias fundamentales existen con el ciclo de vida tradicional en cascada?	10
19. ¿Cuáles son los aspectos fundamentales de la metodología SCRUM?	10
20. En SCRUM ¿qué diferencia hay entre el “product backlog” y el “sprint backlog”?.....	11
21. ¿Cuáles son los roles principales de quienes participan en un proyecto utilizando la metodología SCRUM?	11
3. Proceso de análisis y técnicas	11
1. ¿Qué es un requisito?	11
2. ¿Qué diferencia hay entre los requisitos de usuario y los requisitos de sistema?	12
3. Tipos de requisitos.....	12
4. ¿Qué es un modelo? (en el contexto de la ingeniería del software).....	12
5. Cita y describe brevemente los elementos básicos de los diagramas de casos de uso	12
6. ¿Qué tareas deben llevarse a cabo para obtener un modelo funcional utilizando casos de uso (según Cockburn)?	13
7. ¿Qué aspectos debemos tener en cuenta al especificar los flujos de los casos de uso?.....	13
8. ¿Qué diferencia hay entre los flujos principales y flujos alternativos de los casos de uso?	13
9. ¿Qué técnicas se pueden utilizar para especificar el comportamiento del software? ¿Qué diferencias hay entre ellas?	14
10. Cita y describe brevemente los elementos básicos de los diagramas de clases	14
11. ¿En qué actividades de la metodología Métrica se obtiene un modelo de negocio basado en diagramas de clases? ¿qué se debe hacer detalladamente en dichas actividades?	14
12. ¿Qué técnica se ha utilizado en la asignatura para especificar los requisitos de sistema relativos a la estructura de la información? ¿Qué productos se obtienen? ¿De qué forma deben utilizarse posteriormente esos productos en el diseño?	14

13. ¿Qué notación se utiliza en los diagramas de transición de estados?	14
14. ¿En qué consiste la trazabilidad de requisitos?	15
15. ¿Qué partes debe tener la especificación de un proyecto de desarrollo de software?	15
4. Proceso de diseño	15
1. ¿Qué diferencias hay entre el análisis y el diseño de software?	15
2. ¿Cuáles son los objetivos del diseño? Compáralos con los objetivos del análisis	16
3. Enumera las técnicas aplicables durante la fase de diseño que conozcas. ¿En qué parte del diseño (arquitectura, detallado, datos) se utiliza cada una?.....	16
4. Menciona las técnicas aplicables en el diseño de arquitectura que conozcas. Pon un ejemplo de alguna de ellas.....	16
5. ¿Qué semejanzas y diferencias hay entre los subsistemas de análisis y los subsistemas de diseño?	16
6. ¿Cómo se hace un Diagrama de Despliegue y para qué sirve? Poner un ejemplo	16
7. ¿Con qué objetivo se utiliza la técnica de los Diagramas de Secuencia en el diseño? Pon algún ejemplo de Diagrama de Secuencia	17
8. Describe qué elementos se pueden encontrar en un Diagrama de Secuencia y pon un pequeño ejemplo de uno de estos diagramas	18
9. ¿Qué semejanzas y diferencias hay entre el modelado de datos en el análisis y el diseño de datos? ¿Tienen los mismos objetivos?	18
10. ¿Qué productos se obtienen como resultado del diseño de detalle? Pon algún ejemplo	18
11. ¿Qué es el diseño de datos?	18
12. ¿Qué debe hacerse en el diseño de datos?	19
5. Procesos de pruebas	19
1. ¿Cuáles son los objetivos del proceso de pruebas?	19
2. Define lo que es un caso de prueba y pon un ejemplo	19
3. ¿Qué productos se obtienen como resultado del proceso de pruebas?	19
4. ¿Cómo se utiliza el concepto de cobertura a efectos de probar software? Poner un ejemplo.....	19
5. ¿Existe equivalencia entre la cobertura de sentencias y de decisiones? Si es así, explicar el motivo. Si no, poner un ejemplo que demuestre que no son equivalentes.	20
6. ¿Qué es una técnica de prueba? ¿para qué utilizamos técnicas de prueba?.....	20
7. ¿Qué diferencias hay entre las pruebas unitarias, de integración y de sistema?	20
8. ¿Qué es el modelo en V?	20
9. ¿En qué momento se diseñan y ejecutan las pruebas de aceptación?	21
10. En un proyecto de desarrollo de software ¿se ejecutarán antes las pruebas de componentes o las pruebas de integración? ¿por qué? ¿pueden diseñarse las pruebas de integración antes que las de componentes?	21
11. Describe el proceso que sigue un ingeniero para realizar las pruebas de un sistema software	21
12. ¿En qué consisten las pruebas de regresión?	22
Enero 2017	22
1. ¿Cuáles son los objetivos del análisis de software?	22
2. Define lo que es un incremento en ingeniería del software y lo que significa que se aborda un proyecto con un enfoque incremental.	22
3. Describe cómo se debe especificar un modelo de dominio. En la documentación de un proyecto respecto al modelo de dominio ¿basta con poner un diagrama o es necesario añadir algún contenido/apartado más?	22
4. Explicar en qué consiste la arquitectura cliente-servidor.	23
5. Indicar como se llama, en que consiste y poner un ejemplo de la técnica de diseño en que se transforman las clases a tablas. El ejemplo debe incluir al menos 3 clases y una relación n-m	23
6. Explicar en qué consiste la cobertura de múltiple condición. Poner un ejemplo adecuado.	24
Junio 2018	24
1. ¿Cuáles son los objetivos del Estudio de Viabilidad del Sistema en la metodología Métrica?	24

2. ¿De qué partes consta la documentación de un proyecto de desarrollo de software? Puedes tomar como referencia el reglamento de TFG de la EPIG.	24
3. Detallar en qué consiste el análisis de consistencia.	25
4. En la fase de diseño, describir las actividades relacionadas con el diseño de clases y el diseño físico de datos.	25
5. Indica la relación entre los procesos de desarrollo y los procesos de pruebas.	25
Junio 2015	26
1. Indica cuáles son los procesos de la metodología Métrica relacionados con el desarrollo de sistemas de información y describe brevemente cada uno.	26
2. ¿Qué técnica se ha utilizado en la asignatura para especificar los requisitos de sistema relativos a la estructura de la información? Describe brevemente dicha ficha técnica y pon un pequeño ejemplo	26
3. ¿Qué diferencia existe entre los procesos de desarrollo de software incrementales e iterativos? ¿Es SCRUM incremental o iterativo? Razona la respuesta.	26
4. En qué consiste la validación en el desarrollo de un sistema software.	27
5. En qué consiste la “definición de la arquitectura del sistema”.	27
6. Indica cuál es el objetivo de las pruebas del software, qué es un “fallo” y que se considera “una buena prueba”.	27
Junio 2016	27
1. ¿Qué diferencia hay entre los requisitos de usuario y los requisitos de sistema? ¿Como se especifican ambos tipos de requisitos en un proyecto de desarrollo software?	27
2. ¿En qué consiste la trazabilidad de requisitos? Pon un ejemplo	28
3. Arquitectura de tres capas: definición, ventajas e inconvenientes. Pon un ejemplo con un diagrama de paquetes indicando cada una de las capas	28
4. ¿Que describe el diseño del software? ¿Qué tipos hay?	29
5. Dado el siguiente fragmento de código:	29
Junio 2019	30
1. Describe brevemente las actividades del proceso ASI de la metodología Métrica.	30
2. Qué relación existe entre los elementos de los diagramas de transición de estados y el resto de los modelos que se han utilizado en la asignatura para realizar el análisis detallado de un sistema? Pon un pequeño ejemplo.	30
3. Explicar en qué consiste la definición de la arquitectura del sistema.	30
4. Las asociaciones de un diagrama de clases, cómo se transforman al pasar un diagrama de clases a tablas.	30
Julio 2013	31
1. Las acciones en los flujos de los casos de uso representan:	31
2. ¿Qué es un ciclo de vida incremental? ¿Es incremental el ciclo de vida descrito en la metodología SCRUM? Razona tu respuesta indicando resumidamente las características del ciclo de vida de la metodología SCRUM	31
Enero 2020	31
1. Nombra las actividades del proceso ASI de Métrica e indica con detalle en qué orden deben realizarse, enfatizando cuáles de ellas tiene sentido ejecutar de forma simultánea y cuáles no.	31
2. En el análisis de consistencia y especificación de requisitos ¿en qué consiste concretamente el análisis de consistencia?.....	31
3. Tipos de diseño, indicar en qué consiste cada uno de ellos.	31
4. De acuerdo con el esquema en V que relaciona desarrollo y pruebas ¿Con qué parte del desarrollo se corresponden las pruebas de aceptación y por qué?	31
Mayo 2020	32
1. Enumera las técnicas aplicables en la fase de diseño e indica para cada una de ellas en qué parte del diseño se utilizan.	32
2. ¿Qué objetivos tiene el proceso EVS de la metodología Métrica? ¿y el proceso ASI? Indica las similitudes y diferencias entre ambos.	32

1. Introducción

1. ¿Qué es la ingeniería del software?

La aplicación sistemática de conocimientos científicos y tecnológicos para crear métodos y experiencias para el diseño de un software que responda a las necesidades del usuario y su implementación, así como para realizar pruebas sobre este y su correspondiente documentación. También se trata de la aplicación de un enfoque sistemáticos y cuantificable del desarrollo y operaciones para el mantenimiento de un software.

2. ¿Qué son los procesos de software? Enumera qué procesos genéricos se abordan en un proyecto de desarrollo de software, indicando brevemente en qué consiste cada uno de ellos.

Es un conjunto de actividades y productos obtenidos durante el desarrollo de un sistema software, independientemente de su tamaño y complejidad.

Los procesos genéricos son:

- Análisis: Que hay que hacer.
- Diseño: Como hay que hacerlo.
- Codificación: Hacerlo.
- Prueba: Comprobar si está correcto.
- Mantenimiento: Mejorar, adaptar y corregir.

2. Metodologías de desarrollo de software

1. ¿Cuáles son los procesos principales de Métrica 3? Descríbelos brevemente.

PLANIFICACIÓN DE SISTEMAS DE INFORMACIÓN (Proceso PSI)

Tiene como objetivo la obtención de un marco de referencia para el desarrollo de sistemas de información que responda a los objetivos estratégicos de la organización

DESARROLLO DE SISTEMAS DE INFORMACIÓN

Contiene todas las actividades y tareas que se deben llevar a cabo para desarrollar un sistema, cubriendo desde el análisis de requisitos hasta la instalación del software. Incluye dos partes en el diseño de sistemas: arquitectónico y detallado. También cubre las pruebas unitarias y de integración del sistema. Se subdivide en cinco procesos:

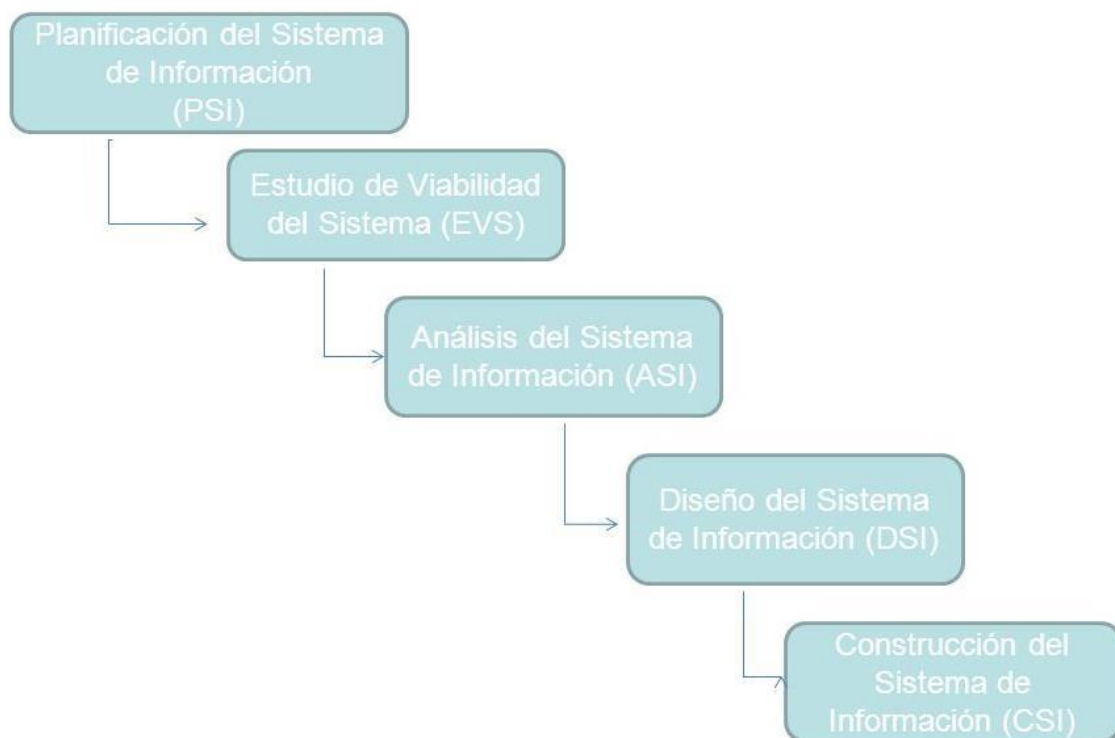
- Estudio de Viabilidad del Sistema (Proceso EVS) → tiene como objetivo analizar un conjunto concreto de necesidades, con la idea de proponer una solución a corto plazo.
- Análisis del Sistema de Información (Proceso ASI) → tiene como objetivo conseguir la especificación detallada del sistema de información, a través de un catálogo de requisitos y una serie de modelos que cubran las necesidades de información de los usuarios para los que se desarrollará el sistema de información.

- Diseño del Sistema de Información (Proceso DSI) → tiene como objetivo la definición de la arquitectura del sistema y del entorno tecnológico que le va a dar soporte, junto con la especificación detallada de los componentes del sistema de información.
- Construcción del Sistema de Información (CSI) → tiene como objetivo la construcción y prueba de los distintos componentes del sistema de información, a partir del conjunto de especificaciones lógicas y físicas del mismo.
- Implantación y Aceptación del Sistema (Proceso IAS) → tiene como objetivo la entrega y aceptación del sistema en su totalidad y llevar a cabo las actividades oportunas para el paso a producción del sistema.

MANTENIMIENTO DE LOS SISTEMAS DE INFORMACION (Proceso MSI)

Tiene como objetivo la obtención de una nueva versión de un sistema de información a partir de las peticiones de mantenimiento que los usuarios realizan con motivo de un problema detectado en el sistema, o por la necesidad de una mejora de este.

2. Realiza un diagrama GANTT que represente los procesos principales de Métrica 3 en un proyecto de desarrollo de software y descríbelos brevemente.



3. ¿Cuál es el objetivo del EVS?

El objetivo principal del Estudio de Viabilidad del Sistema (EVS) es analizar un conjunto de necesidades específicas para proponer una solución a corto plazo para el desarrollo de un sistema de información. Esto incluye el análisis de los requisitos, la evaluación de la viabilidad técnica y económica del proyecto, así como el diseño de una arquitectura de software para satisfacer las necesidades del usuario. El EVS también es útil para identificar posibles errores de diseño antes de que el software sea construido, lo que permite a los desarrolladores ahorrar tiempo y dinero al corregir errores temprano en el proceso de desarrollo.

- EVS 1: Establecimiento del alcance del sistema
- EVS 2: Estudio de la situación actual
- EVS 3: Definición de requisitos del sistema
- EVS 4: Estudio de alternativas de solución
- EVS 5: Valoración de alternativas
- EVS 6: Selección de la solución

4. ¿En qué orden se deben realizar las actividades del proceso EVS?

1. Establecimiento del alcance del sistema.
2. Estudio de la situación actual.
3. Definición de requisitos del sistema.
4. Estudio de alternativas de solución.
5. Valoración de alternativas.
6. Selección de la solución.

5. ¿Qué se hace y qué se obtiene en la actividad EVS 1: establecimiento del alcance del sistema?

Se realiza un análisis de los requisitos del sistema e identifican los objetivos del proyecto. También se establecen los límites del sistema y se definen los procesos y los datos que formarán parte del sistema. Como resultado de esta actividad, se obtiene una descripción completa y precisa del alcance del sistema que se desarrollará.

6. ¿Qué debe estudiarse en el EVS respecto a la situación actual?

EVS 2: Estudio de la situación actual

Conocer cómo se encuentran los sistemas de información al inicio del estudio

- Valoración de la situación actual
- Identificación de usuarios participantes
- Conocimiento y descripción de los sistemas existentes
- Diagnóstico de situación

El objetivo del estudio de la situación actual es conocer cómo se encuentran los sistemas de información al inicio del estudio. Teniendo en cuenta este objetivo, se realiza una valoración de la información existente acerca de los sistemas de información afectados. También se constituye un equipo de trabajo específico para su realización y se identifican los usuarios participantes en el mismo. Se describe cada uno de los sistemas existentes. Como resultado de esta actividad se genera un diagnóstico, estimando la eficiencia de los sistemas de información existentes e identificando los posibles problemas y las mejoras.

7. ¿Qué debe hacerse en el EVS respecto al análisis y valoración de la alternativa de construcción?

EVS 4: Estudio de alternativas

- Preselección de alternativas diferentes
- Descripción de cada una (cobertura de requisitos, cobertura geográfica, paquetes comerciales, entorno tecnológico, costes de licencias y HW, etc.)

EVS 5: Valoración de alternativas

- Estudio de Inversión (viabilidad económica) y de los Riesgos.
- Planificación de cada alternativa. Enfoque adecuado para implantación. Sincronización con otros procesos o proyectos.

Se deben de estudiar todas las alternativas posibles, describiendo en cada una que requisitos cubre y la cobertura que ofrece. También se ha de estudiar la inversión económica y los riesgos que producirá cada una de ella, así como realizar modificaciones en ellas. Por último, se debe de seleccionar una entre todas para implementarla.

8. ¿Cuál es el objetivo del ASI?

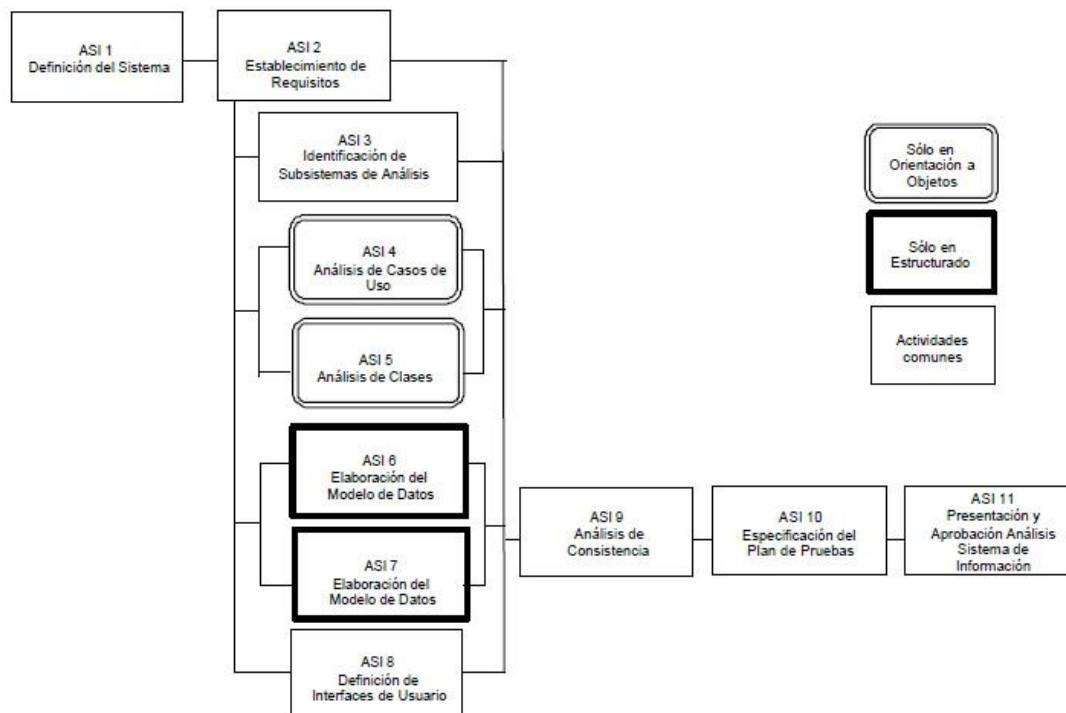
El objetivo del ASI (Análisis del Sistema de Información) es la obtención de una especificación detallada del Sistema de Información (la alternativa seleccionada) a través del catálogo de requisitos y de los modelos del sistema, de una base para el posterior diseño del sistema y del marco metodológico de Métrica v3 que cubre técnicas estructuradas (clásicas) y orientadas a objetos, pero en una estructura común.

9. Actividades del proceso ASI

- **Definición del Sistema (ASI 1)** → Tiene como objetivo efectuar una descripción del sistema, delimitando su alcance, estableciendo las interfaces con otros sistemas e identificando a los usuarios representativos.
- **Establecimiento de Requisitos (ASI 2)** → Se lleva a cabo la definición, análisis y validación de los requisitos a partir de la información facilitada por el usuario, completándose el catálogo de requisitos.
- **Identificación de Subsistemas de Análisis (ASI 3)** → Tiene como objetivo facilitar el análisis del sistema de información llevando a cabo la descomposición del sistema en subsistemas.
- **Análisis de Casos de Uso (ASI 4)** → Tiene como objetivo identificar las clases cuyos objetos son necesarios para realizar un caso de uso y describir su comportamiento mediante la interacción dichos objetos.
- **Análisis de Clases (ASI 5)** → Tiene como objetivo describir cada una de las clases que ha surgido, identificando las responsabilidades que tienen asociadas, sus atributos, y las relaciones entre ellas
- **Elaboración del Modelo de Datos (ASI 6)** → Tiene como objetivo identificar las necesidades de información de cada uno de los procesos que conforman el sistema de información, con el fin de obtener un modelo de datos que contemple todas las entidades, relaciones, atributos y reglas de negocio necesarias para dar respuesta a dichas necesidades.
- **Elaboración del Modelo de Procesos (ASI 7)** → Tiene como objetivo analizar las necesidades del usuario para establecer el conjunto de procesos que conforma el sistema de información.
- **Definición de Interfaces de Usuario (ASI 8)** → Se especifican las interfaces entre el sistema y el usuario: formatos de pantallas, diálogos, e informes.
- **Análisis de Consistencia y Especificación de Requisitos (ASI 9)** → Tiene como objetivo garantizar la calidad de los distintos modelos generados en el proceso ASI, y asegurar que los usuarios y los analistas tienen el mismo concepto del sistema.

- **Especificación del Plan de Pruebas (ASI 10)** → Se inicia la definición del plan de pruebas, el cual sirve como guía para la realización de las pruebas, y permite verificar que el sistema de información cumple las necesidades establecidas por el usuario, con las debidas garantías de calidad.
- **Aprobación del Análisis del Sistema de Información (ASI 11)** → Se realiza la presentación del análisis del sistema de información al Comité de Dirección, para la aprobación final del mismo.

10. Realiza un diagrama GANTT que represente las actividades del proceso ASI y descríbelas brevemente



Actividades del proceso ASI descritas en el [ejercicio anterior](#)

11. Cita y describe brevemente los modelos utilizados en el proceso ASI

- **Modelo de datos:** Solo se elabora en el caso de Análisis estructurado. Con este modelo identificamos las necesidades de información de cada uno de los procesos que conforman el sistema de información con el fin de obtener un modelo de datos que contemple todas las entidades, relaciones y atributos para dar respuesta a dichas necesidades.
- **Modelo de procesos:** También se elabora solo en el caso de Análisis estructurado, con este modelo analizamos las necesidades del usuario para establecer un conjunto de procesos que conforma el sistema de información. Para ello se descomponen los procesos en varios niveles de abstracción donde cada nivel proporciona una visión más detallada del proceso definido en el proceso anterior.

12. ¿Qué productos se obtienen en la actividad del ASI denominada “Definición de interfaces de usuario”?

- Principios generales de la interfaz.
- Catálogo de perfiles de usuario.
- Descomposición funcional en diálogos.
- Catálogo de controles y elementos de diseño de interfaz de pantalla.
- Formatos individuales de interfaz de pantalla.
- Modelo de navegación de interfaz de pantalla.
- Formatos de impresión.
- Prototipo de interfaz interactiva.
- Prototipo de interfaz de impresión.

13. ¿Cuáles son los objetivos de la actividad del ASI denominada Análisis de Consistencia y Especificación de Requisitos?

- **Calidad formal de los modelos:** analizar/repasar los modelos realizados en busca de fallos o errores
- Confrontar los modelos entre ellos para determinar inconsistencias o lagunas.
- Trazabilidad de los requisitos a través del modelo y mostrar prototipo al usuario.
- **Reunir el resultado del análisis en un documento:** Especificación de Requisitos del Software (ERS)

El objetivo de esta actividad es garantizar la calidad de los distintos modelos generados en el proceso ASI, y asegurar que los usuarios y los analistas tienen el mismo concepto del sistema.

14. ¿Qué actividades del proceso ASI de la metodología Métrica 3 no sea bordan en las prácticas de la asignatura?

- Elaboración del Modelo de Datos
- Elaboración del Modelo de Procesos

15. Resume los objetivos y productos más importantes del proceso de Diseño del Sistema de Información de la metodología Métrica 3

Los objetivos fundamentales del proceso de diseño son:

- La arquitectura del sistema
- El detalle de los subsistemas
- Las estructuras de datos necesarias

Productos:

- Las especificaciones de construcción (DSI 8)
- El diseño de módulos y de datos
- El plan de pruebas (DSI 10)

16. ¿En qué proceso de la metodología Métrica se codifica el software? Menciona los productos fundamentales de ese proceso, algunas de sus actividades y con qué proceso continúa la metodología

En el proceso de Construcción del Sistema de Información (CSI).

El producto fundamental obtenido en este proceso es el código del sistema ya probado, además también se obtendrán los manuales de usuario. En este proceso se genera el código de los componentes del Sistema de Información, se desarrollan todos los procedimientos de operación y seguridad y se elaboran todos los manuales de usuario final y de explotación con el objetivo de asegurar el correcto funcionamiento del Sistema para su posterior implantación.

Algunas de las actividades del proceso CSI son:

- Preparación del Entorno de Generación y Construcción
- Generación del Código de los Componentes y Procedimientos
- Ejecución de las Pruebas Unitarias
- Ejecución de las pruebas de Integración
- Ejecución de las Pruebas del Sistema
- Elaboración de los Manuales de Usuario
- Definición de la Formación de Usuarios Finales
- Construcción Componentes y Procedimientos de migración y Carga Inicial de Datos
- Aprobación del Sistema de Información

La metodología Métrica continúa con el proceso de Implantación y Aceptación del Sistema (IAS).

17. ¿Cuáles son los objetivos principales del proceso de Implantación y Aceptación del Sistema en la metodología Métrica?

Los objetivos principales del proceso de Implantación y Aceptación del Sistema (IAS) son la entrega y aceptación del sistema en su totalidad y el paso a producción del mismo.

18. ¿Qué es un ciclo de vida incremental? ¿Qué diferencias fundamentales existen con el ciclo de vida tradicional en cascada?

Con ciclo de vida nos referimos a cómo se ordenan y coordinan los diferentes procesos. Un ciclo de vida incremental permite la entrega de versiones parciales a medida que se va construyendo el producto final. Es decir, a medida que cada incremento definido llega a su etapa de operación y mantenimiento. Cada secuencia lineal produce un producto operacional con cada incremento de la misma forma que progresa el tiempo. Los primeros incrementos se pueden implementar con menos recursos.

La diferencia es que en este modelo no es necesario terminar una etapa para comenzar con la siguiente, mientras que en el ciclo de cascada sí.

19. ¿Cuáles son los aspectos fundamentales de la metodología SCRUM?

Los aspectos fundamentales de la metodología SCRUM son los siguientes:

- Marco de trabajo iterativo e incremental

- Estructura el desarrollo en Sprint
- Al final de cada sprint los productos realizados deben estar integrados, probados y potencialmente entregables
- Hincapié en pequeños pasos de desarrollo, inspeccionar resultados y adaptar

20. En SCRUM ¿qué diferencia hay entre el “product backlog” y el “sprint backlog”?

El product backlog es un documento de alto nivel para todo el proyecto. Contiene descripciones genéricas de todos los requisitos, funcionalidades deseables, etc. Priorizadas según su retorno sobre la inversión. Es el qué va a ser construido. Es abierto y solo puede ser modificado por el product owner. Contiene estimaciones realizadas a grandes rasgos.

El sprint backlog es un documento detallado donde se describe el cómo el equipo va a implementar los requisitos durante el siguiente sprint. Las tareas se dividen en horas con ninguna tarea de duración superior a 16 horas. Las tareas en el sprint backlog nunca son asignadas, son tomadas por los miembros del equipo del modo que les parezca oportuno.

21. ¿Cuáles son los roles principales de quienes participan en un proyecto utilizando la metodología SCRUM?

Roles principales:

- **Product Owner:** El Product Owner representa la voz del cliente. Se asegura de que el equipo Scrum trabaje de forma adecuada desde la perspectiva del negocio. El Product Owner escribe historias de usuario, las prioriza, y las coloca en el Product Backlog.
- **ScrumMaster (o Facilitador):** El Scrum es facilitado por un ScrumMaster, cuyo trabajo primario es eliminar los obstáculos que impiden que el equipo alcance el objetivo del sprint. El ScrumMaster no es el líder del equipo (porque ellos se auto-organizan), sino que actúa como una protección entre el equipo y cualquier influencia que le distraiga. El ScrumMaster se asegura de que el proceso Scrum se utiliza como es debido. El ScrumMaster es el que hace que las reglas se cumplan.
- **Equipo de desarrollo:** El equipo tiene la responsabilidad de entregar el producto. Un pequeño equipo de 3 a 9 personas con las habilidades transversales necesarias para realizar el trabajo (análisis, diseño, desarrollo, pruebas, documentación, etc.).

3. Proceso de análisis y técnicas

1. ¿Qué es un requisito?

Un requisito es una “condición o capacidad que necesita el usuario para resolver un problema o conseguir un objetivo determinado”. También se aplica a las condiciones que debe cumplir o poseer un sistema o uno de sus componentes para satisfacer un contrato, una norma o una especificación.

Puede verse como:

- Una declaración abstracta de alto nivel de un servicio que el sistema debe proporcionar
- Una definición matemática detallada y formal de una función del sistema.

Los requisitos cumplen una doble función:

- Son una oferta de contrato → Abiertos a la interpretación
- Son el contrato en sí mismo → Deben definirse de forma detallada

2. ¿Qué diferencia hay entre los requisitos de usuario y los requisitos de sistema?

Representan dos niveles diferentes de especificación de los requisitos. Los requisitos de usuario están orientados a definir los servicios necesarios que el sistema ha de proporcionar para las partes interesadas y los usuarios en un contexto previamente definido. Mientras, los requisitos de sistema son una “transformación” de los requisitos de usuario en una visión más detallada del producto que ha de satisfacer dichos servicios.

- **Sistema:** Muestra, lista algo, calcula. EL sistema muestra todas las inscripciones.
- **Usuario:** inscripción para atletas de distintos tipos.

3. Tipos de requisitos

Requisitos funcionales (RF)

Definición de los servicios que el sistema debe proporcionar, cómo debe reaccionar a una entrada particular y cómo se debe comportar ante situaciones particulares. Servicios y funciones que tendrá el software (lo que hará)

Requisitos no funcionales (RNF)

Restricciones que afectan a los servicios o funciones del sistema, tales como restricciones de tiempo, sobre el proceso de desarrollo, estándares, etc.

4. ¿Qué es un modelo? (en el contexto de la ingeniería del software)

Un modelo es una representación de un objeto, sistema o idea. El propósito de los modelos es ayudarnos a explicar, entender o mejorar un sistema. Un modelo de un objeto puede ser una réplica exacta de éste o una abstracción de las propiedades dominantes del objeto.

5. Cita y describe brevemente los elementos básicos de los diagramas de casos de uso

Actores:

Representan un tipo de usuario del sistema, que no siempre son usuarios (humanos)

- Unidades organizativas
- Otros sistemas

Relaciones:

- Asociación
- Relaciones entre casos de uso y actores
- El actor interactúa (o participa) con el sistema para llevar a cabo el caso de uso
- Se representan por una línea no dirigida y sin etiqueta

Caso de uso:

Secuencia de acciones que describen una tarea o proceso realizado por un actor con el sistema

Cada uno de los casos de uso debe especificarse textualmente mediante uno o varios flujos:

- Flujo principal
- Flujos alternativos

Prototipo:

Conjunto de operaciones de una clase o paquete visibles desde otras clases o paquetes.

6. ¿Qué tareas deben llevarse a cabo para obtener un modelo funcional utilizando casos de uso (según Cockburn)?

- Elaborar una lista de actores y objetivos.
- Elaborar los principales escenarios que se llevan a cabo con éxito.
- Listar las diferentes condiciones que hacen que algo falle.
- Tratar con esas condiciones.

7. ¿Qué aspectos debemos tener en cuenta al especificar los flujos de los casos de uso?

- Las acciones de los flujos deben ser sencillas, es decir, utilizar un enfoque jerárquico, explicar acciones complejas utilizando acciones más sencillas.
- Las acciones representan:
 - Algo que hace un actor
 - Algo que hace el sistema
- Analizar el orden en que se ejecutarán las acciones, numerarlas.
- Los flujos deben ser completos: indicar todas las acciones necesarias para llevar a cabo el caso de uso.
- Centrarse en lo esencial, evitando referencias a diseño detallado de pantallas (hacer clic en un botón).
- Al finalizar un flujo alternativo, el caso de uso puede volver al flujo principal o puede terminar, indicarlo.
- Especificar en qué momentos el usuario puede cancelar el caso de uso. Algunas cancelaciones requerirán la ejecución de acciones compensatorias.
- Si es necesario indicar precondiciones y postcondiciones (el estado en que debe encontrarse el sistema antes/después de ejecutar un caso de uso).

8. ¿Qué diferencia hay entre los flujos principales y flujos alternativos de los casos de uso?

En el flujo principal se especifica lo que sucederá normalmente al ejecutar el caso de uso.

Mientras que en los flujos alternativos se especifican las variantes o los tratamientos excepcionales. Al finalizar un flujo alternativo, el caso de uso puede volver al flujo principal o puede terminarlo.

9. ¿Qué técnicas se pueden utilizar para especificar el comportamiento del software?
¿Qué diferencias hay entre ellas?

- **Diagramas de clase:** Propone un conjunto de clases (Objetos) y las diferentes relaciones que tienen entre sí para poder explicar el comportamiento del software.
- **Diagramas de transición de estados:** Propone un conjunto de estados por los que pasa una clase determinada, comenzando en un estado inicial y acabando en uno final, para explicar el comportamiento del software.

10. Cita y describe brevemente los elementos básicos de los diagramas de clases

- **Clases:** Describen un conjunto de objetos con propiedades y comportamientos comunes.
- **Relaciones:** Enlaces entre los distintos elementos de los diagramas.
- **Interfaces:** Conjunto de operaciones de una clase o paquete visibles desde otras clases o paquetes.

11. ¿En qué actividades de la metodología Métrica se obtiene un modelo de negocio basado en diagramas de clases? ¿qué se debe hacer detalladamente en dichas actividades?

- **ASI 3: Identificación de Subsistemas de Análisis** → Se descompone el sistema en subsistemas para facilitar el análisis. Se divide en base de procesos similares o que trabajan sobre los mismos datos.
- **ASI 4: Análisis de los casos de uso** → Se identifican las clases cuyos objetos se necesitan para realizar un caso de uso y se describe su comportamiento mediante interacción de dichos objetos. Se hace para cada uno de los casos de uso.
- **ASI 5: Análisis de clases** → Se describir cada una de las clases. Se analizan las asociaciones para determinar su tipo y las clases en busca de generalizaciones o especializaciones (herencia), y se añaden a las asociaciones su cardinalidad correcta, nombres de rol, etc.

12. ¿Qué técnica se ha utilizado en la asignatura para especificar los requisitos de sistema relativos a la estructura de la información? ¿Qué productos se obtienen? ¿De qué forma deben utilizarse posteriormente esos productos en el diseño?

En la actividad EVS 1 (establecimiento del alcance del sistema) se estudia el alcance de la necesidad planteada por el cliente o usuario. Se inicia el estudio de la solicitud y se identifica el alcance del sistema y del EVS, identificando las unidades organizativas afectadas estableciendo su estructura.

13. ¿Qué notación se utiliza en los diagramas de transición de estados?

Estado:

Un estado se representa como un rectángulo con las esquinas redondeadas. El nombre del estado se coloca dentro del rectángulo y debe ser único en el diagrama. Si se repite algún nombre, se asume que simboliza el mismo estado.

Las acciones y actividades descritas como respuesta a eventos que no producen un cambio de estado se representan dentro del rectángulo con el formato: nombre-evento (parámetros) [condición] /acción

El estado inicial se representa con un pequeño círculo relleno, y el estado final como un pequeño círculo relleno con una circunferencia que lo rodea.

Transición:

Una transición se representa con una flecha continua que une dos estados y que se dirige al estado al que cambia el componente. Junto a ella se coloca una etiqueta que debe contener al menos el nombre del evento que provoca la transición. Según el nivel de detalle, puede presentar otros elementos con el formato siguiente: nombre-evento (parámetros) [condición] /acción

14. ¿En qué consiste la trazabilidad de requisitos?

Es la habilidad de describir y seguir la vida de un requisito desde su origen, pasando por su desarrollo y especificación hasta su despliegue. Ha de hacerse en ambas direcciones, hacia adelante y hacia atrás.

Nos permite identificar el origen de cada requisito y seguir cada cambio que se realice sobre el mismo. Pero no sólo eso, al trazar los requisitos con otros artefactos (pruebas, casos de uso, planes de proyecto, etc.) será posible responder a los cambios en el proyecto de forma más controlada y con más información. Podremos anteponernos a lo que un cambio puede significar.

15. ¿Qué partes debe tener la especificación de un proyecto de desarrollo de software?

- **Especificación de software:** Se debe definir la funcionalidad y restricciones operacionales que debe cumplir el software.
- **Diseño e Implementación:** Se diseña y construye el software de acuerdo con la especificación.
- **Validación:** El software debe validarse, para asegurar que cumpla con lo que quiere el cliente.
- **Evolución:** El software debe evolucionar, para adaptarse a las necesidades del cliente.

4. Proceso de diseño

1. ¿Qué diferencias hay entre el análisis y el diseño de software?

La diferencia fundamental es que el objetivo de análisis de software es la obtención de una especificación detallada del sistema de información que satisfaga las necesidades de información de los usuarios y sirva de base para el posterior diseño del sistema, mientras que el objetivo del Diseño de software es la definición de la arquitectura del sistema y del entorno tecnológico que le va a dar soporte, junto con la especificación detallada de los componentes del sistema de información.

2. ¿Cuáles son los objetivos del diseño? Compáralos con los objetivos del análisis

- El objetivo del diseño es la definición de la arquitectura del sistema y del entorno tecnológico que le dará soporte, además de una especificación de los componentes del sistema de información.
- Los objetivos del análisis, en cambio, es la obtención de una especificación detallada del sistema de información que satisfaga las necesidades de los usuarios y que sirva de base para el diseño.

3. Enumera las técnicas aplicables durante la fase de diseño que conozcas. ¿En qué parte del diseño (arquitectura, detallado, datos) se utiliza cada una?

- **Diagramas de Despliegue** → Arquitectura
- **Diagramas de Paquetes** → Arquitectura
- **Reglas de Transformación** → Datos
- **Optimización** → Detallado

4. Menciona las técnicas aplicables en el diseño de arquitectura que conozcas. Pon un ejemplo de alguna de ellas

- **Diagramas de Despliegue** → Parte Arquitectura
- **Diagramas de Paquetes** → Parte Arquitectura
- **Diagramas de nodos** → Parte Arquitectura
- **Diagramas interacción** → Parte Diseño
- **Reglas de Transformación** → Parte Datos
- **Optimización** → Parte Detallado

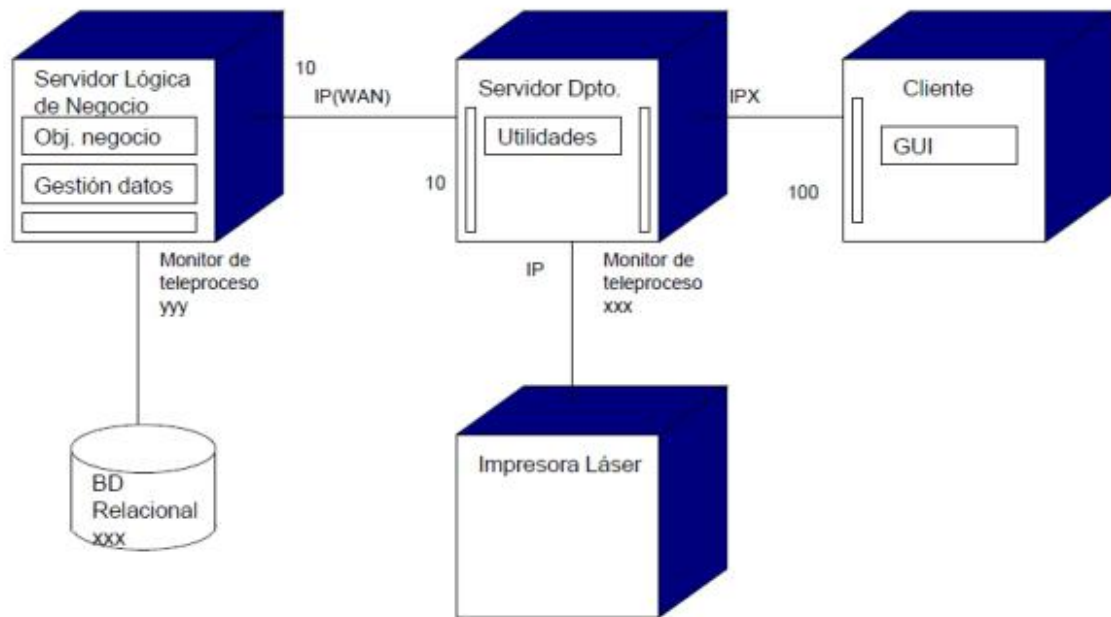
5. ¿Qué semejanzas y diferencias hay entre los subsistemas de análisis y los subsistemas de diseño?

El objetivo principal del análisis es recopilar información y definir los requisitos, mientras que el objetivo principal del diseño es crear un sistema para satisfacer los requisitos definidos. análisis requiere una comprensión profunda del problema y una evaluación detallada de las soluciones posibles, mientras que el diseño requiere una comprensión de los requisitos y una visión de cómo se pueden satisfacer. El análisis también involucra la validación de los requisitos del sistema

6. ¿Cómo se hace un Diagrama de Despliegue y para qué sirve? Poner un ejemplo

Sirve para representar las particiones físicas del sistema de información y los interfaces existentes entre ellas.

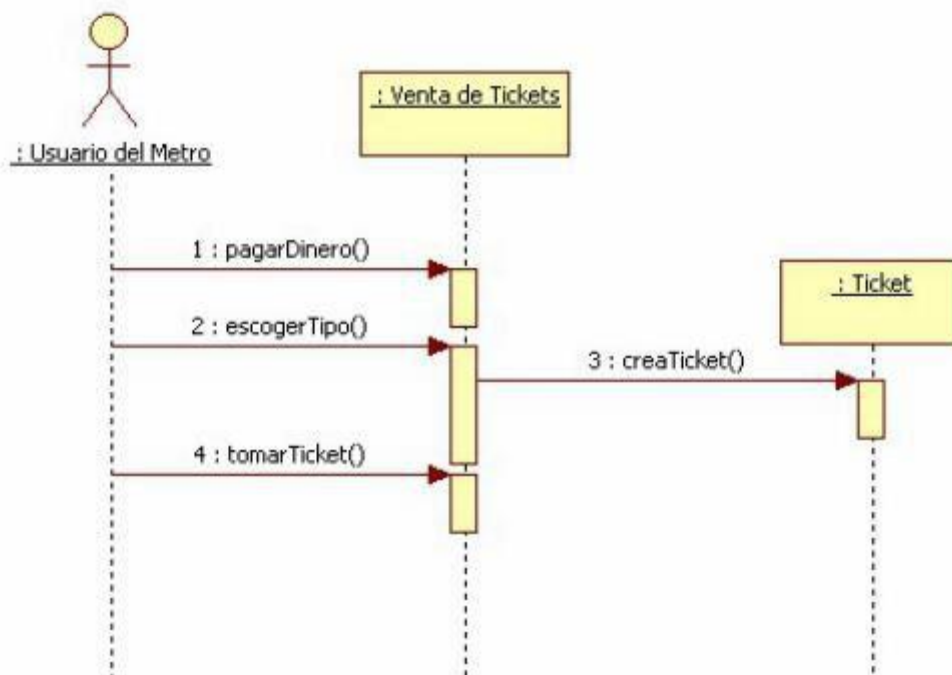
Hay que representar los nodos como cubos y las comunicaciones entre estos como líneas. Con esto reflejamos los protocolos de las comunicaciones y la cantidad de nodos existentes.



7. ¿Con qué objetivo se utiliza la técnica de los Diagramas de Secuencia en el diseño?
Pon algún ejemplo de Diagrama de Secuencia

El diagrama de secuencia es un tipo de diagrama de interacción cuyo objetivo es describir el comportamiento dinámico del sistema de información haciendo énfasis en la secuencia de los mensajes intercambiados por los objetos.

su objetivo es describir como interactuaran los diferentes componentes principales del sistema para cumplir el objetivo de cada uso o también para describir cómo interactúan los elementos internos de cada componente para lograr el resultado necesario para cada mensaje entrante.



8. Describe qué elementos se pueden encontrar en un Diagrama de Secuencia y pon un pequeño ejemplo de uno de estos diagramas

- **Objeto:** se representa como una línea vertical discontinua, llamada línea de vida, con un rectángulo de encabezado con el nombre del objeto en su interior. También se puede incluir a continuación el nombre de la clase, separando ambos por dos puntos.
- **Línea de vida:** la línea de vida de un objeto puede desplegarse en dos o más líneas para mostrar los diferentes flujos de mensajes que puede intercambiar un objeto, dependiendo de alguna condición.
- **Foco de control o activación:** se representa como un rectángulo delgado superpuesto a la línea de vida del objeto. Su largo dependerá de la duración de la acción. La parte superior del rectángulo indica el inicio de una acción ejecutada por el objeto y la parte inferior su finalización.
- **Mensaje:** se representa como una flecha horizontal entre las líneas de vida de los objetos que intercambian el mensaje. La flecha va desde el objeto que envía el mensaje al que lo recibe.

(Ejemplo en el [diagrama de la pregunta anterior](#))

9. ¿Qué semejanzas y diferencias hay entre el modelado de datos en el análisis y el diseño de datos? ¿Tienen los mismos objetivos?

Ambos tienen el objetivo de organizar y representar los datos de forma óptima.

- **Modelado de datos:** Se enfoca en la comprensión de los datos, los datos se organizan para facilitar la comprensión de estos y se enfoca en la construcción de modelos narrativos para explicar el comportamiento de los datos.
- **Diseño de datos:** Se enfoca en la manipulación y el uso de datos, los datos se organizan para mejorar la eficiencia y flexibilidad del uso de estos y se enfoca en la construcción de modelos predictivos.

10. ¿Qué productos se obtienen como resultado del diseño de detalle? Pon algún ejemplo

Los productos incluyen diagramas de clases, diagramas de flujo de control, diagramas de estado, descripciones de requisitos funcionales y no funcionales, listas de tareas y pruebas. Por ejemplo, una lista de tareas es un documento que contiene una lista detallada de todas las tareas que los desarrolladores deben completar para implementar una solución de software. Esta lista de tareas puede incluir tareas relacionadas con la codificación, pruebas, documentación, etc.

11. ¿Qué es el diseño de datos?

El diseño de datos transforma el modelo de dominio de información que se crea durante el análisis en las estructuras de datos que se necesitarán para implementar el software.

Los objetos de datos y las relaciones definidas en el diagrama relación entidad y el contenido de datos detallado que se representa en el diccionario de datos proporcionan la base de la actividad del diseño de datos.

12. ¿Qué debe hacerse en el diseño de datos?

Desarrollar y revisar representaciones del flujo y contenido de datos, identificar estructuras de datos, considerar estructuras de datos alternativas y evaluar el impacto de la modernización de datos sobre el diseño del software.

Un ejemplo de esto puede ser transformar diagramas de clases a tablas, incorporación de claves primarias, ajenas y alternativas.

5. Procesos de pruebas

1. ¿Cuáles son los objetivos del proceso de pruebas?

El proceso de pruebas consiste en ejecutar el software con la intención de descubrir fallos. También se puede definir como un proceso que consiste en la ejecución de un programa bajo ciertos datos de entrada (casos de prueba), para posteriormente comparar las salidas obtenidas con las deseadas (función para la cual fue diseñado). Tienen el objetivo de mejorar la calidad del software. Dan confianza en el producto.

2. Define lo que es un caso de prueba y pon un ejemplo

Un caso de prueba (CP) es un conjunto de entradas, condiciones de ejecución y resultados esperados desarrollados para un objetivo particular.

Caso práctico: carrito de la compra en tienda virtual de libros.

Objetivo: Probar la acción de introducir un nuevo ejemplar de un libro en el carrito.

Condiciones de la prueba:

El catálogo contendrá, al menos, el siguiente libro: 6-332-72934-0 con precio de 70€. El stock del libro será de 3 unidades.

El carrito estará vacío en el momento de introducir el ejemplar.

Entradas: ISBN = 6-332-72934-0, Unidades = 2.

Salida deseada: El libro se introduce correctamente en el carrito, el precio actual del carrito pasa a ser de 140 € y el stock del libro se actualiza a 1 unidad.

3. ¿Qué productos se obtienen como resultado del proceso de pruebas?

Como consecuencia de la realización de pruebas, se detectarán errores o fallos. Cuando una prueba hace visible un error, es necesario corregirlo, buscando y eliminando el defecto que lo provoca.

4. ¿Cómo se utiliza el concepto de cobertura a efectos de probar software? Poner un ejemplo

Como no es factible ejercitar el software de todas las formas posibles se deberá de diseñar un conjunto razonable de casos con un criterio de prueba. Esto puede dar cobertura de sentencias, de decisiones, de condiciones y de múltiple condición.

5. ¿Existe equivalencia entre la cobertura de sentencias y de decisiones? Si es así, explicar el motivo. Si no, poner un ejemplo que demuestre que no son equivalentes.

Sí, existe equivalencia entre la cobertura de sentencias y de decisiones. La cobertura de sentencias consiste en que toda sentencia ejecutable en un programa debe ser invocada al menos una vez. En la cobertura de decisiones se diseñan casos de prueba que ejerciten las vertientes verdadera y falsa de cada decisión en un programa. Con este criterio, se asegura la prueba completa de las estructuras de control del programa siempre y cuando las decisiones sean simples, esto es, no estén formadas por la combinación lógica de condiciones. Por definición, se asegura la cobertura de sentencias. Para el ejemplo (a OR b) con dos casos de prueba (a=true, b=false) y (a=false, b=false) se asegura la cobertura de decisión.

6. ¿Qué es una técnica de prueba? ¿para qué utilizamos técnicas de prueba?

Una técnica de prueba o criterio de prueba es un procedimiento por el que se decide qué casos de prueba se deben ejecutar sobre el software. Se realiza maximizando la probabilidad de encontrar fallos y cumpliendo límites de tiempo y coste.

Tienen el objetivo de seleccionar buenos casos de prueba, esto es, casos que tengan una probabilidad alta de descubrir un error.

7. ¿Qué diferencias hay entre las pruebas unitarias, de integración y de sistema?

- **Pruebas unitarias o pruebas modulares:** constituyen el primer paso para detectar errores en el código, pues se centran en la menor unidad de diseño del software: el módulo (por ejemplo, un método de una clase o una clase). El objetivo principal de estas pruebas es detectar errores en cada uno de los módulos del software al ser ejecutados independientemente del resto de componentes.
- **Pruebas de integración:** parten de los componentes individuales previamente probados y tienen como objetivo descubrir errores que se pueden producir en la interacción entre los módulos.
- **Pruebas de sistema:** sus objetivos son la detección de errores en el producto ya integrado. En ellas se ejercitan condiciones que anteriormente habría sido difícil comprobar, como las interfaces entre subsistemas, la comunicación con el entorno o el cumplimiento de los requisitos no funcionales.

8. ¿Qué es el modelo en V?

Es una representación gráfica del ciclo de vida del desarrollo de sistemas. En él se resumen las principales medidas que deben adoptarse en relación con las prestaciones correspondientes en el marco del sistema informático de validación.

Es un proceso que representa la secuencia de pasos en el desarrollo del ciclo de vida de un proyecto. Se describen las actividades y resultados que deben producirse durante el desarrollo del producto. El lado izquierdo de la V representa la descomposición de las necesidades, y la creación de las especificaciones del sistema. El lado derecho de la V representa la integración de las piezas y su verificación. V significa «Verificación y validación». Es muy similar al modelo de la cascada clásico ya que es muy rígido y contiene una gran cantidad de iteraciones.

9. ¿En qué momento se diseñan y ejecutan las pruebas de aceptación?

Las pruebas se realizan para que el cliente certifique que el sistema es válido para él. La planificación detallada de estas pruebas debe haberse realizado en etapas tempranas del desarrollo del proyecto, con el objetivo de utilizar los resultados como indicador de la validez del proyecto: si se ejecutan las pruebas documentadas a satisfacción del cliente, el producto se considera correcto, y por tanto adecuado para su puesta en producción.

El objetivo de las pruebas de aceptación es validar que un sistema cumple con el funcionamiento esperado y permitir al usuario de dicho sistema que determine su aceptación, desde el punto de vista de su funcionalidad y rendimiento. Las pruebas de aceptación son definidas por el usuario del sistema y preparadas por el equipo de desarrollo, aunque la ejecución y aprobación final corresponden al usuario. Estas pruebas van dirigidas a comprobar que el sistema cumple los requisitos de funcionamiento esperado, recogidos en el catálogo de requisitos y en los criterios de aceptación del sistema de información, y conseguir así la aceptación final del sistema por parte del usuario.

10. En un proyecto de desarrollo de software ¿se ejecutarán antes las pruebas de componentes o las pruebas de integración? ¿por qué? ¿pueden diseñarse las pruebas de integración antes que las de componentes?

Se ejecutarán antes las pruebas de componentes que las pruebas de integración ya que las pruebas de integración parten de los componentes individuales previamente probados y tienen como objetivo descubrir errores que se pueden producir en la interacción entre los módulos. En teoría, la combinación de componentes válidos debería dar como resultado software en el que no se detectan errores, pero en la práctica hay múltiples ocasiones en las que las pruebas de unidad no detectan errores que sí se desvelan al ejecutar pruebas sobre el software integrado.

En la integración incremental se combina el siguiente componente que se debe probar con el conjunto de componentes que ya están probados y se va incrementando progresivamente el número de componentes a probar. En la integración no incremental se prueba cada

11. Describe el proceso que sigue un ingeniero para realizar las pruebas de un sistema software

En la metodología Métrica v3 se comienza la planificación de los procesos de prueba en el Análisis del Sistema de Información: se empieza por definir el alcance de las pruebas, los entornos en los que se ejecutarán y planteando pruebas de aceptación. Siguiendo la metodología, en el diseño se completará la especificación técnica de los entornos de prueba y se obtendrán casos de prueba desde un punto de vista de caja negra, pues aún no se ha construido el código. Conforme se avanza en el desarrollo del proyecto, se empiezan a construir módulos de código en el proceso de Construcción del Sistema de Información (CSI). En ese momento, se podrán incluir casos de prueba de caja blanca, independientemente de que se añadan también otras pruebas de caja negra, y comenzar su ejecución, concretamente en las actividades denominadas CSI 3, CSI 4 y CSI 5. En concreto, CSI 3 comienza mientras se está generando el código de los componentes que, según son validados, pueden acoplarse a otros para, a continuación, aplicar pruebas de integración. El software completo debe someterse, para terminar el proceso de construcción, a pruebas de sistema (CSI 5). Tras la

construcción, Métrica establece que debe realizarse el proceso de Implantación y Aceptación del Software (IAS), en el que se instala el software en el entorno de operación y, por tanto, se ejecutan las pruebas de implantación (IAS 5). Finalmente, la ejecución de las pruebas de aceptación se realiza justo antes de la aprobación del sistema y el paso a producción, en la actividad sexta del proceso IAS.

12. ¿En qué consisten las pruebas de regresión?

El objetivo de las pruebas de regresión es eliminar el efecto onda, es decir, comprobar que los cambios sobre un componente de un sistema de información no introducen un comportamiento no deseado o errores adicionales en otros componentes no modificados. Las pruebas de regresión se deben llevar a cabo cada vez que se hace un cambio en el sistema, tanto para corregir un error como para realizar una mejora. No es suficiente probar sólo los componentes modificados o añadidos, o las funciones que en ellos se realizan, sino que también es necesario controlar que las modificaciones no produzcan efectos negativos sobre el mismo u otro componente.

Enero 2017

1. ¿Cuáles son los objetivos del análisis de software?

Los principales objetivos del análisis de software en ingeniería de software son identificar los requisitos del software, validar la arquitectura y el diseño del software, evaluar el rendimiento y la calidad del producto, documentar el proceso de desarrollo del software, prevenir errores en el desarrollo del software, mejorar la calidad del producto y asegurar que el software cumpla con los requisitos del cliente. Estas tareas pueden ser realizadas mediante técnicas como el análisis estructurado, el análisis orientado a objetos, el análisis de componentes, el análisis de casos de uso, el análisis de pruebas, el análisis de estándares, y el análisis de seguridad.

2. Define lo que es un incremento en ingeniería del software y lo que significa que es aborda un proyecto con un enfoque incremental.

Un incremento es versión de software funcional del sistema, la cual en cada uno de estos incorpora nuevas funcionalidades. Abordar un proyecto con un enfoque incremental consiste en dividir el desarrollo en incrementos priorizando los requisitos más relevantes.

3. Describe cómo se debe especificar un modelo de dominio. En la documentación de un proyecto respecto al modelo de dominio ¿basta con poner un diagrama o es necesario añadir algún contenido/apartado más?

La especificación de un modelo de dominio se realiza a través de un diagrama de clases que se utilizará posteriormente para el diseño de la base de datos en la que se reflejará la información relevante del sistema de información de datos.

Las clases describen un conjunto de objetos con propiedades y comportamientos comunes enlazadas a través de relaciones. Las clases están en formato un UML específico de la asignatura en la que las clases no tienen relaciones.

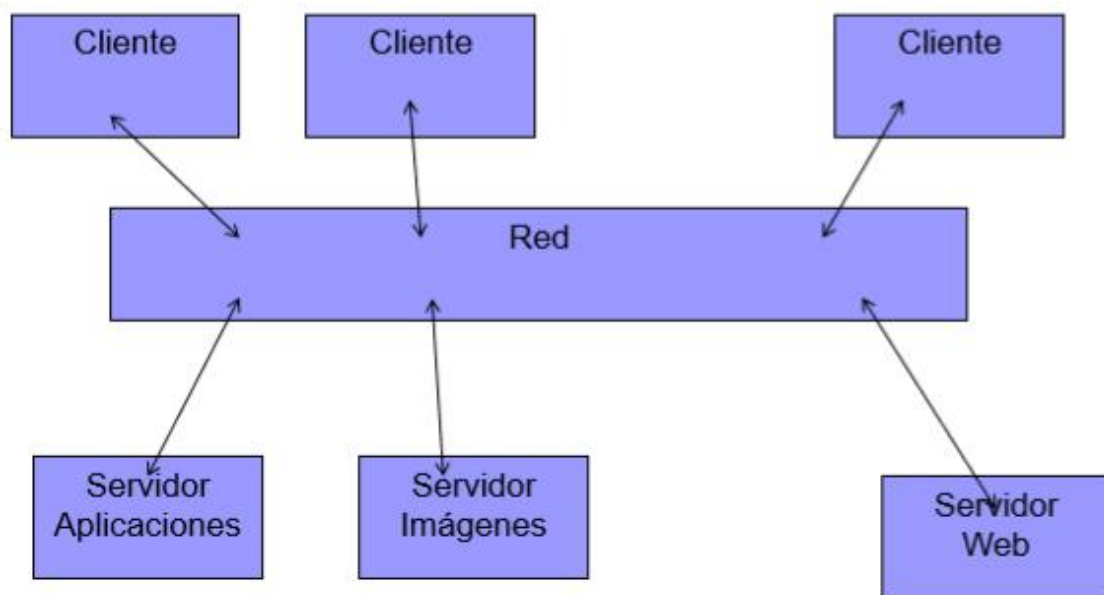
Relaciones: reflejan los enlaces entre los distintos elementos de los diagramas.

Son de dos tipos:

- **Asociaciones:** relaciones estructurales. Estas pueden tener nombres, roles, y una multiplicidad para indicar el número de objetos involucrados en la asociación.
- **Generalización:** herencias

Los modelos de dominio deben ir siempre acompañados de un diccionario de datos para explicar o definir los diferentes atributos que necesiten alguna explicación y/o aclaración

4. Explicar en qué consiste la arquitectura cliente-servidor.



Cliente y servidor están conectados mediante una red, el servidor debe de estar siempre activo, se entiende como servidor la maquina o maquinas que den servicio a nuestro sistema pueden ser (De aplicación, de imágenes o web). Cumple estas características:

- Distribución de datos real
- Uso de sistemas en red correcta
- Fácil ampliación
- Cada subsistema puede usar modelos de datos diferentes → ineficiente
- Gestor en cada servidor → redundancia
- Difícil controlar que servicios están disponibles

5. Indicar como se llama, en qué consiste y poner un ejemplo de la técnica de diseño en que se transforman las clases a tablas. El ejemplo debe incluir al menos 3 clases y una relación n-m

La técnica de diseño en que se transforman las clases a tablas se conoce como Normalización de Entidades Relacionales. Esta técnica consiste en transformar las clases y sus atributos en tablas y relaciones, respetando las restricciones de integridad referencial y la normalización para evitar la redundancia de datos. Por ejemplo, si se tienen las clases Estudiante, Estado y Curso, se pueden transformar en las siguientes tablas y relaciones:

- **Tabla Estudiante:**
 - Estudiante ID
 - Nombre
 - Apellido
 - Estado ID
- **Tabla Estado:**
 - Estado ID
 - Nombre
- **Tabla Curso:**
 - Curso ID
 - Nombre
- **Relación Estudiante-Curso (n-m):**
 - Estudiante ID
 - Curso ID

6. Explicar en qué consiste la cobertura de múltiple condición. Poner un ejemplo adecuado.

Esta cobertura requiere la evaluación de todas las posibles combinaciones de valores de cada condición en cada decisión. Para el ejemplo (a OR b), se consigue esta cobertura con los casos (a=true, b=true), (a=true, b=false), (a=false, b=true) y (a=false, b=false). Obviamente esta medida de cobertura es la más completa de las anteriores, sin embargo, es poco práctica debido al gran número de casos de prueba que es necesario diseñar para su cumplimiento.

Junio 2018

1. ¿Cuáles son los objetivos del Estudio de Viabilidad del Sistema en la metodología Métrica?

- Analizar las necesidades del cliente/usuario para proponer solución a corto plazo
- Comprender por qué el cliente/usuario necesita ese cambio en su sistema de información actual, delimitar lo que se precisa y plantear una solución

Da como resultado la definición de uno o varios proyectos a través de un establecimiento del alcance del sistema, estudio de la situación actual y estudio y valoración de alternativas, y la selección de una solución (puede no haber solución).

2. ¿De qué partes consta la documentación de un proyecto de desarrollo de software? Puedes tomar como referencia el reglamento de TFG de la EPIG.

1. **Memoria**
2. **Presupuesto del desarrollo efectuado**
3. **Documentos técnicos**
 - a. Requisitos de usuario o pliego de condiciones
 - i. Descripción general, ámbito y alcance del proyecto
 - ii. Lista de usuarios participantes
 - iii. Descripción del sistema actual
 - iv. Catálogo de requisitos del sistema y prioridades

- v. Análisis de alternativas
 - b. Análisis de requisitos del sistema
 - i. Especificación funcional
 - ii. Interfaces de usuario
 - iii. Modelo de dominio
 - iv. Modelo dinámico
 - v. Análisis de consistencia
 - c. Diseño (de arquitectura y de detalle)
 - i. Diseño de la arquitectura del sistema
 - ii. Diseño detallado de los subsistemas de diseño
 - iii. Diseño físico de datos
 - d. Pruebas (diseño y resultados de ejecución)
 - i. Plan de pruebas
 - ii. Diseño de pruebas
 - iii. Informe ejecución de pruebas
- 4. Manual de usuario y del instalador (si es necesario)**
- 5. Aplicación instalable**

3. Detallar en qué consiste el análisis de consistencia.

En la confrontación (comparar) de distintos diseños del sistema para detectar ambigüedades, falta de información, o incoherencias. Ej: Casos de uso vs Requisitos funcionales

Comprobar que las técnicas usadas en los diseños sean válidas y correctas

Más información en: [Pregunta 13, punto 4](#)

4. En la fase de diseño, describir las actividades relacionadas con el diseño de clases y el diseño físico de datos.

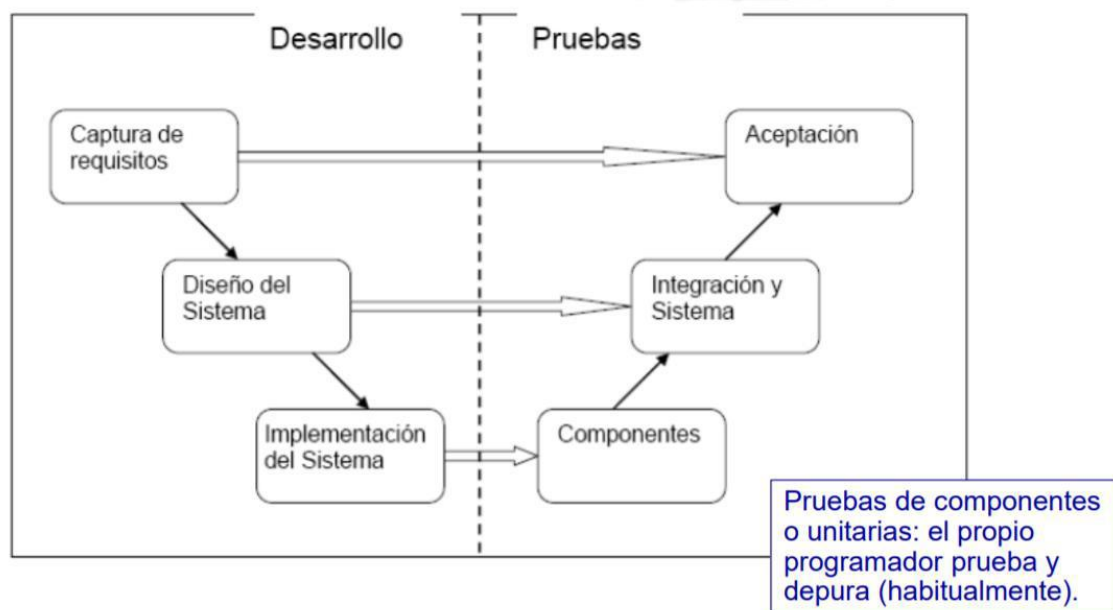
- Identificación de las clases que intervienen en los Casos de Uso (a nivel de Diseño)
- Partiendo de los escenarios, detallarlos teniendo en cuenta que ahora se conoce el entorno tecnológico.
- Identificación de las clases que completan el modelo de dominio
- Las clases pueden verse modificadas por el entorno tecnológico.
- Conocido el entorno de desarrollo se incorporan los atributos necesarios y las operaciones que den cobertura a las necesidades de cada clase.
- Desarrollo del modelo de datos a nivel físico (base de datos), determinación de cómo se convertirán las clases a tablas, incorporación de claves primarias, ajenas, alternativas.

5. Indica la relación entre los procesos de desarrollo y los procesos de pruebas.

Es una representación gráfica del ciclo de vida del desarrollo de sistemas. En él se resumen las principales medidas que deben adoptarse en relación con las prestaciones correspondientes en el marco del sistema informático de validación.

Es un proceso que representa la secuencia de pasos en el desarrollo del ciclo de vida de un proyecto. Se describen las actividades y resultados que deben producirse durante el desarrollo del producto. El lado izquierdo de la V representa la descomposición de las necesidades, y la creación de las especificaciones del sistema. El lado derecho de la V representa la integración

de las piezas y su verificación. V significa «Verificación y validación». Es muy similar al modelo de la cascada clásico ya que es muy rígido y contiene una gran cantidad de iteraciones.



Junio 2015

1. Indica cuáles son los procesos de la metodología Métrica relacionados con el desarrollo de sistemas de información y describe brevemente cada uno.

Respuesta en [Pregunta 1, punto 2](#) (Solo la parte de Desarrollo de sistemas de información)

2. ¿Qué técnica se ha utilizado en la asignatura para especificar los requisitos de sistema relativos a la estructura de la información? Describe brevemente dicha ficha técnica y pon un pequeño ejemplo

En la actividad EVS 1 (establecimiento del alcance del sistema) se estudia el alcance de la necesidad planteada por el cliente o usuario. Se inicia el estudio de la solicitud y se identifica el alcance del sistema y del EVS, identificando las unidades organizativas afectadas estableciendo su estructura. (Sacado de [Pregunta 12, apartado 3](#))

Por ejemplo, si se trata de desarrollar un sistema de gestión de recursos humanos, los requisitos incluirían el almacenamiento de información sobre los empleados, la gestión de las nóminas, la gestión de los beneficios, la gestión de los horarios de trabajo, etc. Los componentes del software incluirían una interfaz de usuario, una base de datos, un algoritmo de gestión de recursos humanos, etc.

3. ¿Qué diferencia existe entre los procesos de desarrollo de software incrementales e iterativos? ¿Es SCRUM incremental o iterativo? Razona la respuesta.

La principal diferencia entre los procesos de desarrollo de software incrementales e iterativos es que los procesos de desarrollo incremental se centran en la entrega de una versión completa del software al final de cada ciclo de desarrollo, mientras que los procesos de

desarrollo iterativo se centran en la entrega de una versión parcial del software en cada ciclo de desarrollo.

SCRUM es un proceso de desarrollo de software iterativo. Esto se debe a que SCRUM se centra en la entrega de una versión parcial del software en cada ciclo de desarrollo, lo que le permite al equipo recibir retroalimentación temprana para realizar ajustes necesarios. Además, la estimación de tiempo de SCRUM se basa en entregas parciales y no en una versión completa del software.

4. En qué consiste la validación en el desarrollo de un sistema software.

Consiste en asegurarse de que el sistema cumple los requisitos previamente establecidos por el cliente. Esto implica verificar que el sistema funciona de acuerdo con el diseño, que cumple con los requisitos de usuario y que es seguro y confiable. Esta validación se lleva a cabo mediante una serie de pruebas, como pruebas unitarias, pruebas de integración, pruebas de aceptación y pruebas de seguridad. Estas pruebas se deben realizar en cada etapa del desarrollo del sistema para asegurar que el sistema cumple con los requisitos previamente establecidos.

5. En qué consiste la “definición de la arquitectura del sistema”.

- Particionar el sistema desde el punto de vista físico Qué nodos se usarán Qué comunicaciones deben establecerse entre los nodos
- Asignar subsistemas a nodos
- Diseñar los subsistemas de soporte
- Identificar mecanismos genéricos de diseño (ejemplo: validaciones mediante controles)

6. Indica cuál es el objetivo de las pruebas del software, qué es un “fallo” y que se considera “una buena prueba”.

El objetivo principal de las pruebas del software es asegurarse de que el software cumple con los requisitos establecidos y que funciona de manera correcta y confiable. Un “fallo” es un resultado de la prueba que indica que el software no cumple con los requisitos establecidos. Una buena “prueba” es aquella que identifica los fallos del software y ayuda a determinar cómo solucionarlos. Además, una buena prueba debe ser específica, detallada, precisa y cuantificable.

Junio 2016

1. ¿Qué diferencia hay entre los requisitos de usuario y los requisitos de sistema? ¿Como se especifican ambos tipos de requisitos en un proyecto de desarrollo software?

Representan dos niveles diferentes de especificación de los requisitos. Los requisitos de usuario están orientados a definir los servicios necesarios que el sistema ha de proporcionar para las partes interesadas y los usuarios en un contexto previamente definido. Mientras, los requisitos de sistema son una “transformación” de los requisitos de usuario en una visión más detallada del producto que ha de satisfacer dichos servicios. (Sacado de [Pregunta 2, punto 3](#))

- **Requisitos de usuario:** Se especifican generalmente mediante un documento de requisitos de usuario, que contiene todos los requisitos funcionales y no funcionales esperados por el usuario.
- **Requisitos de sistema:** Se especifican generalmente mediante un documento de requisitos de sistema, que contiene todos los requisitos técnicos que el sistema debe cumplir para implementar los requisitos de usuario.

2. ¿En qué consiste la trazabilidad de requisitos? Pon un ejemplo

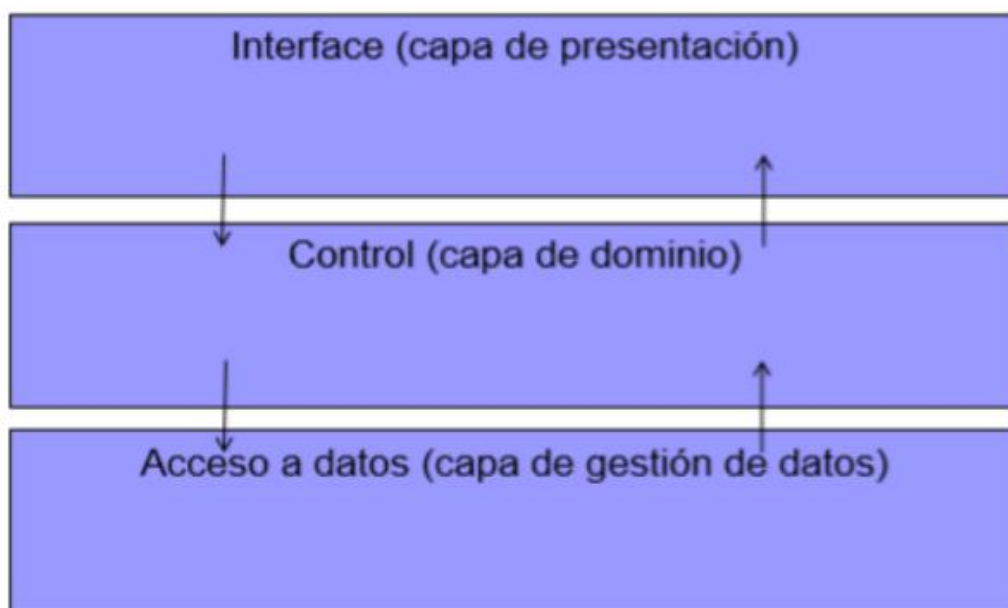
Es la habilidad de describir y seguir la vida de un requisito desde su origen, pasando por su desarrollo y especificación hasta su despliegue. Ha de hacerse en ambas direcciones, hacia adelante y hacia atrás.

Nos permite identificar el origen de cada requisito y seguir cada cambio que se realice sobre el mismo. Pero no sólo eso, al trazar los requisitos con otros artefactos (pruebas, casos de uso, planes de proyecto, etc.) será posible responder a los cambios en el proyecto de forma más controlada y con más información. Podremos antepoñernos a lo que un cambio puede significar. (Sacado de [Pregunta 14, punto 3](#))

Un ejemplo de trazabilidad de requisitos puede ser el desarrollo de un sistema de información para una empresa. En este proyecto, se deben especificar los requisitos tanto del usuario como del sistema. Por ejemplo, se especificarían los requisitos de usuario como la funcionalidad, la usabilidad, la seguridad, etc., y los requisitos de sistema como el hardware, el software, la tecnología, etc. Esta información se usará para construir el sistema de información y asegurar que se cumplan los requisitos. La herramienta de trazabilidad de requisitos se usaría para rastrear los requisitos desde su definición hasta su implementación.

3. Arquitectura de tres capas: definición, ventajas e inconvenientes. Pon un ejemplo con un diagrama de paquetes indicando cada una de las capas

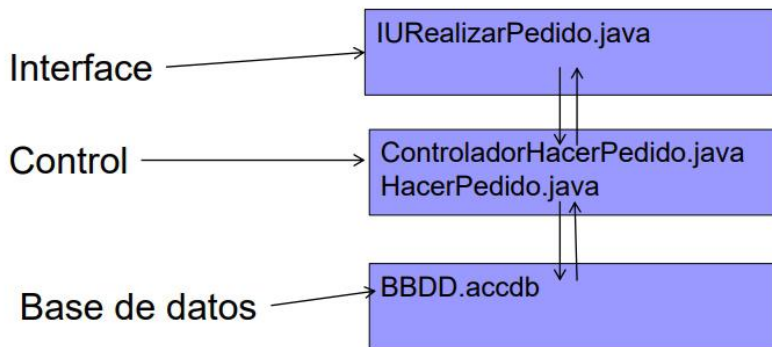
Arquitectura de tres capas (habitual)



- **Definición:** División en tres partes (capas) interface, control y acceso a datos que se encargan de gestionar de forma independiente aislando la lógica de cada una.
- **Ventajas:** Cambios fáciles en cada nivel con pocos efectos sobre el resto. Aísla la lógica de la aplicación convirtiéndola en una capa intermedia. Permite distribuir las capas por nodos físicos.
- **Implementación física:**
 - La capa de presentación puede implementarse en el nodo cliente.
 - La capa de negocio puede implementarse en un nodo servidor de aplicaciones.
 - La capa de datos en el nodo servidor de DB.
- **Ejemplo:**

Implementación Lógica-Ejemplo

- Sistema: Máquinas Vending
 - Caso de Uso: “Realizar pedido”



4. ¿Que describe el diseño del software? ¿Qué tipos hay?

- **Describe:**
 - La arquitectura del software (organización en componentes)
 - Los interfaces entre componentes.
 - El detalle de cada componente para poder implementarlo
- **Tipos de diseño:**
 - Arquitectura: Describe la estructura y organización a alto nivel. (subsistemas, componentes y sus relaciones). Es el primer paso en el diseño y su resultado es la arquitectura del sistema.
 - Detallado: Describe cada componente y su comportamiento, permite su construcción.

5. Dado el siguiente fragmento de código:

```

accion vender_elemento (elemento, cantidad, urgente)
    sí (elemento no está en lista productos)
        error (Elemento no disponible)
  
```

```

sino
    existencias = disponibilidad producto (elemento)
    sí(cantidad < existencias)
        actualizar stock (elemento, -cantidad);
sino
    prevision - llegada producto (elemento)
    pendiente = encargos(elemento)
    sí (cantidad < existencias + previsión) && (urgente)
        mensaje ("Se puede dejar encargado");

```

Finaccion

Obtén casos de prueba para garantizar la cobertura de condiciones.

Sin respuesta.

Junio 2019

1. Describe brevemente las actividades del proceso ASI de la metodología Métrica.

Respuesta en [Pregunta 9, punto 2](#)

2. Qué relación existe entre los elementos de los diagramas de transición de estados y el resto de los modelos que se han utilizado en la asignatura para realizar el análisis detallado de un sistema? Pon un pequeño ejemplo.

Todos sirven para modelar el sistema, con la diferencia que el diagrama de transición de estados muestra una actividad (caso de uso) en concreto exponiendo todos sus estados de forma gráfica por ejemplo un diagrama de actividad muestra de un caso de uso el comportamiento general con un enfoque ordenado en el tiempo. Aunque los dos se basen en el mismo caso de uso modelan partes distintas.

3. Explicar en qué consiste la definición de la arquitectura del sistema.

Respuesta en [Pregunta 5, junio 2015](#)

4. Las asociaciones de un diagrama de clases, cómo se transforman al pasar un diagrama de clases a tablas.

- **M a N**: Se transforman en una tabla. La clave primaria se obtiene concatenando los identificadores de las clases relacionadas. Los atributos de la relación se convierten en columnas de la nueva tabla.
- **1 a N**: Existen varias posibilidades: Añadir el identificados de la clase del extremo con cardinalidad 1 a la clase del otro extremo como clave ajena. Una tabla al igual que si tuviera multiplicidad M a N

- **1 a 1:** Caso particular de 1 a N

¿Qué son las pruebas de aceptación? ¿Cuándo se definen?

Respuesta en [Pregunta 9, punto 5](#)

Julio 2013

1. Las acciones en los flujos de los casos de uso representan:

- a. Algo que hace el software.
- b. Algo que hacen los sistemas externos.
- c. **Algo que hace un actor. (Respuesta correcta)**

2. ¿Qué es un ciclo de vida incremental? ¿Es incremental el ciclo de vida descrito en la metodología SCRUM? Razona tu respuesta indicando resumidamente las características del ciclo de vida de la metodología SCRUM.

Respuesta en [Pregunta 3, junio 2015](#)

Enero 2020

1. Nombra las actividades del proceso ASI de Métrica e indica con detalle en qué orden deben realizarse, enfatizando cuáles de ellas tiene sentido ejecutar de forma simultánea y cuáles no.

Respuesta en [Pregunta 10, punto 2](#)

2. En el análisis de consistencia y especificación de requisitos ¿en qué consiste concretamente el análisis de consistencia?

Tiene como objetivo garantizar la calidad de los distintos modelos generados en el proceso ASI, y asegurar que los usuarios y los analistas tienen el mismo concepto del sistema. (Sacado de [Pregunta 9, punto 2](#)) Esto implica verificar que los requisitos sean lógicamente coherentes entre sí, que sean claros y consistentes, y que no contengan ambigüedades o redundancias. Esta verificación se realiza mediante la comparación de los requisitos entre sí y su confrontación con la documentación existente.

3. Tipos de diseño, indicar en qué consiste cada uno de ellos.

Respuesta en [Pregunta 4, Junio 2016](#)

4. De acuerdo con el esquema en V que relaciona desarrollo y pruebas ¿Con qué parte del desarrollo se corresponden las pruebas de aceptación y por qué?

Captura de requisitos con aceptación, se deben aceptar todos los requisitos por parte del cliente.

Estas pruebas se realizan para que el cliente certifique que el sistema es válido para él. La planificación detallada de estas pruebas debe haberse realizado en etapas tempranas del

desarrollo del proyecto, con el objetivo de utilizar los resultados como indicador de la validez del proyecto: si se ejecutan las pruebas documentadas a satisfacción del cliente, el producto se considera correcto, y por tanto adecuado para su puesta en producción.

Mayo 2020

1. Enumera las técnicas aplicables en la fase de diseño e indica para cada una de ellas en qué parte del diseño se utilizan.

Respuesta en [Pregunta 3, punto 4](#)

2. ¿Qué objetivos tiene el proceso EVS de la metodología Métrica? ¿y el proceso ASI? Indica las similitudes y diferencias entre ambos.

El objetivo del EVS (Estudio de Viabilidad del Sistema) es analizar las necesidades del cliente/usuario (requisitos) para proponer una solución a corto plazo.

El objetivo del ASI (Análisis del Sistema de Información) es la obtención de una especificación detallada del Sistema de Información (la alternativa seleccionada) a través del catálogo de requisitos y de los modelos del sistema, de una base para el posterior diseño del sistema y del marco metodológico de Métrica v3 que cubre técnicas estructuradas (clásicas) y orientadas a objetos, pero en una estructura común.

El EVS es un análisis superficial de sistema con la finalidad de obtener una solución, el así se centra en desarrollar dicha solución.