

Instrucciones

- Cada pregunta múltiple opción respondida correctamente tiene un valor de 4,762 puntos.
- Cada pregunta múltiple opción respondida incorrectamente resta 1,587 puntos.
- Ante dos opciones correctas en una pregunta, se debe seleccionar la opción más completa.
- La evaluación es de carácter individual y la duración es de dos horas.
- El puntaje total es de 100 puntos y se aprueba con 60 o más puntos.

1. Usted es el líder de un equipo de pruebas que trabaja en conjunto con un equipo de desarrollo y se encargan de verificar el software que producen, que es una aplicación móvil para el escaneo de imágenes y reconocimiento de texto (OCR). El equipo trabaja con cuatro ambientes: el ambiente de desarrollo, el ambiente de pruebas, el ambiente de pre-producción, y el ambiente de producción. Ambos equipos (testing y desarrollo) tienen acceso a todos los ambientes. Dadas las siguientes afirmaciones:
- I. Para realizar pruebas unitarias se puede utilizar el ambiente de desarrollo.
 - II. Como en las pruebas de integración quien las ejecuta es el equipo de desarrollo, necesariamente se deben ejecutar en el ambiente de desarrollo.
 - III. Para las pruebas de performance además del ambiente de pre-producción podría ser necesario utilizar el ambiente de producción, cuidando de no impactar de forma significativa a los usuarios en producción.
 - IV. El ambiente de producción podría no utilizarse en ningún nivel de prueba.
 - V. Para las pruebas de aceptación, es necesario usar el ambiente de producción.

Seleccione la opción correcta:

- a) Sólo las afirmaciones I, II, y V son correctas.
 - b) Sólo las afirmaciones II, III y IV son correctas.
 - c) **Sólo las afirmaciones I, III y IV son correctas.**
 - d) Todas las afirmaciones son correctas.
2. Como líder del equipo de pruebas (de la pregunta anterior) Ud. debe planificar el diseño y ejecución de los distintos tipos y niveles de pruebas a realizar. Seleccione la opción correcta sobre la planificación de estas pruebas
- a) Es necesario coordinar la planificación de las pruebas con la planificación del desarrollo, ya que al tener un equipo de pruebas independiente, todos los niveles y tipos de prueba los debe realizar este equipo. Por lo tanto, debe coordinar las actividades de prueba con desarrollo para no retrasar ni entorpecer el avance del proyecto.
 - b) Al haber un equipo de pruebas independiente al equipo de desarrollo, la planificación de las pruebas debe seguir un proceso en cascada ya que en los procesos ágiles el equipo es uno solo y realiza tanto desarrollo como pruebas.
 - c) **La priorización de las pruebas puede tener en cuenta los siguientes aspectos: el impacto de las fallas en el negocio, la criticidad de las funcionalidades para el negocio, fechas comprometidas con el cliente/usuarios, dependencia de funcionalidades/pruebas, entre otros.**
 - d) Siempre se deben ejecutar primero las pruebas unitarias, luego las de integración, luego las de sistema y por último las de aceptación. Bajo ningún concepto se puede saltar un nivel de prueba.

3. Las pruebas de software dinámicas pueden realizarse con diferentes propósitos, que se describen en las siguientes afirmaciones:
- I. Para provocar fallas.
 - II. Para demostrar la ausencia de defectos.
 - III. Para medir la calidad interna del producto.
 - IV. Para medir la calidad externa del producto.
 - V. Para detectar malas prácticas de programación.
 - VI. Para detectar problemas de performance.
 - VII. Para verificar el adecuado cumplimiento de los requisitos

¿Cuál (o cuáles) de ellos serían válidos para este tipo de pruebas (dinámicas)? Seleccione la opción correcta:

- a) Sólo las afirmaciones I, III, IV, V y VII son correctas.
 - b) **Sólo las afirmaciones I, IV, VI y VII son correctas.**
 - c) Sólo las afirmaciones II, III y VI son correctas.
 - d) Sólo las afirmaciones I, IV y VI son correctas.
4. Sobre la gestión de la configuración, seleccione la opción **INCORRECTA**:
- a) Es posible y recomendable incluir en el Plan de SCM cómo se resolverán las situaciones en las que debemos saltar los procedimientos definidos para la gestión de la configuración (o sea las excepciones).
 - b) El alcance y el formalismo de la gestión de la configuración deberían adaptarse a cada proyecto y su contexto.
 - c) En proyectos grandes resulta importante contar con herramientas que apoyen la gestión de la configuración.
 - d) **El responsable de SCM es quién resuelve las solicitudes de cambio.**
5. Respecto a la estructura de desglose de trabajo (WBS), en las dos visiones vistas en el curso se cumple que
- a) Al descomponer el nivel en más detalle mejora la capacidad de planificar, gestionar y controlar. Por eso siempre hay que descomponer una actividad todo lo que sea posible.
 - b) El WBS debe estar siempre balanceado.
 - c) **El nivel de detalle al que se descomponga un WBS varía según el tamaño y la complejidad del proyecto y el conocimiento que se tenga de las actividades.**
 - d) No es necesario que los componentes del nivel inferior sean todos los necesarios y suficientes para completar el nivel superior.
6. En cuanto a la técnica de estimación Delphi:
- a) **Una ventaja de esta técnica es que se pueden tener varias opiniones sin que haya influencias entre los estimadores al no encontrarse en una reunión inicial.**
 - b) Si los resultados convergen después de una o más rondas de estimación, no es necesario hacer una reunión.
 - c) Si después de algunas rondas de estimación los resultados aún no convergen, se puede hacer una nueva ronda de estimaciones unas semanas después.
 - d) Si luego de tres o cuatro rondas de estimación los resultados aún no convergen, se hace una reunión cuyo propósito es conseguir la convergencia de las estimaciones.

7. En cuanto al camino crítico de un proyecto:
- Será aquel camino en el grafo de actividades que comprende las actividades de mayor dificultad técnica, por lo que es el camino de mayor duración.
 - Una vez establecido, se gestiona el cronograma en base a él, ya que no va a cambiar durante el transcurso del proyecto.
 - En la medida que las tareas que están en caminos no críticos se atrasen, pueden pasar a formar parte de un nuevo camino crítico.
 - Puede ir cambiando a lo largo del proyecto, pero siempre es único.
8. Entre las técnicas para acortar el cronograma, se puede utilizar la técnica de *crashing*:
- Esta consiste en intentar comprimir la duración de la actividad, ya sea incorporando más recursos o trabajando horas extra o poniendo incentivos monetarios como premio por entregas antes del plazo.
 - En cuantas más tareas se utilice esta técnica, más efectiva será y se obtendrá una mayor reducción de la duración del proyecto.
 - Si la técnica de crashing se utiliza solo en las actividades del camino crítico, igualmente se obtendrá siempre una reducción en la duración del proyecto.
 - Todas las anteriores son correctas.
9. Que el conjunto de los requisitos sea completo quiere decir:
- que todos los requisitos que el usuario necesita están definidos.
 - que en los requisitos no hay conflictos o definiciones contradictorias.
 - que todos los requisitos especificados son necesarios.
 - que todos los requisitos son verificables mediante un procedimiento viable, medible y cuyo costo se puede pagar.
10. Una de las técnicas de obtención de requisitos son los cuestionarios. El enfoque más adecuado para plantear las respuestas que se piden en un cuestionario es
- La opción múltiple
 - El valor en escala
 - El orden relativo
 - Dependerá de la situación
11. Validar los requisitos consiste en
- Corroborar que el software sea correcto respecto a los requisitos especificados.
 - Determinar si los requisitos relevados son adecuados a las necesidades del cliente.
 - Verificar propiedades de correctitud, completitud y consistencia de la especificación de requisitos.
 - Comprobar la usabilidad del software construido.

12. Seleccione la respuesta correcta

- a) El incremento, de los últimos años, de la complejidad de los sistemas de software que se construyen es un factor más que influye en los defectos (severidad, cantidad, etc.) que los desarrolladores introducen en el software durante el desarrollo. Por eso, para los proyectos más complejos de software de la actualidad se prefiere el modelo en cascada. Esto es para evitar el desconocimiento de cómo voy a desarrollar el sistema.
- b) (a) y no existen al día de hoy procesos de desarrollo de software genéricos, en el sentido que aplican a cualquier contexto y proyecto, debido a que, entre otras cosas, diferentes tipos de software pueden requerir diferentes procesos de desarrollo de software.
- c) (a) y actualmente existen procesos de desarrollo de software generales (que aplican a cualquier contexto y proyecto) debido al avance y mejoras en los últimos años de los métodos ágiles.
- d) a), b) y c) son INCORRECTAS.

13. Una empresa es contratada para desarrollar un cierto producto de software. Luego de trabajar en recolectar al menos una idea general de los requisitos, los líderes del proyecto de desarrollo de software estiman el tamaño del proyecto y el esfuerzo. Dado el tamaño del equipo de desarrollo estiman que el proyecto para la construcción del producto de software pedido llevará alrededor de un año y medio de trabajo. La empresa contratante, expresa que no tiene todos los requisitos claros y que le gustaría abordar en etapas la construcción del software si esto fuera posible. También le gustaría contar con liberaciones parciales del producto para ir viendo cómo funciona y poder pedir cambios si fuera necesario.

- a) Dado que el trabajo va a durar un año y medio, se debería utilizar un método en espiral, ya que al durar ese tiempo van a haber muchos cambios en los requisitos durante el desarrollo.
- b) Parece razonable abordar este proyecto con algún proceso iterativo e incremental con liberaciones parciales al cliente.
- c) Se agrega la siguiente información al contexto presentado en el enunciado de la pregunta: "El equipo de desarrollo tiene experiencia en proyectos llevados adelante con Scrum." Con esta nueva información queda claro que para este proyecto lo indicado es utilizar Scrum.
- d) Dos de las opciones anteriores son correctas.

14. Seleccione la afirmación correcta:

- a) Los procesos iterativos e incrementales deberían liberar incrementos (versiones) al cliente al menos una vez por mes.
- b) Los roles, en la especificación de un proceso de desarrollo, normalmente definen las habilidades y las responsabilidades que deben poder llevar adelante las personas de los equipos de desarrollo. Los roles definen también cómo se hace cada actividad del proceso de desarrollo de software.
- c) El modelo CMMI, por sus características, puede ser utilizado para la mejora de procesos de desarrollo de software (incluso procesos ágiles).
- d) El desarrollo con Scrum debe ser siempre con la práctica "cliente en el lugar". De otra forma, no se está haciendo Scrum sino una variante de Scrum.

15. Seleccione la afirmación correcta:

- a) Cuando se usan procesos iterativos e incrementales en un proyecto, no se realizan liberaciones parciales al cliente. Sin embargo, estos procesos promueven las liberaciones internas (para algunos incrementos) para que el equipo de desarrollo pueda realizar pruebas de sistema.
- b) Scrum define el rol de Scrum Master, entre otros roles.
- c) **b) y Scrum propone revisar en cada *Sprint* la forma en la cual el equipo de desarrollo trabajó durante el *Sprint* y reflexionar sobre cómo se podría haber trabajado mejor.**
- d) c) y los procesos ágiles se centran en la arquitectura de software. En las primeras iteraciones se define y en las siguientes la arquitectura es refinada. Esto, por ejemplo, en Scrum se llama refinamiento sucesivo.

16. Sobre la construcción de software, seleccione la afirmación correcta:

- a) La planificación de la construcción incluye, entre otras cosas, cómo van a ser construidas e integradas las componentes de software.
- b) (a) y la construcción es una etapa del desarrollo de software que normalmente no se realiza en la práctica profesional (es decir, en la industria de software) ya que se pasa directo a la programación luego del diseño.
- c) La construcción efectiva y eficiente que se ha logrado mostrar a través de los últimos años y que sigue vigente es mediante el uso del lenguaje de programación Java.
- d) **a) y el reuso es una actividad que comprende, entre otras cosas, utilizar (reutilizar) software ya existente como parte del desarrollo de software que se está realizando.**

17. De las siguientes figuras, ¿cuál o cuáles podrían representar un diagrama de componentes?:

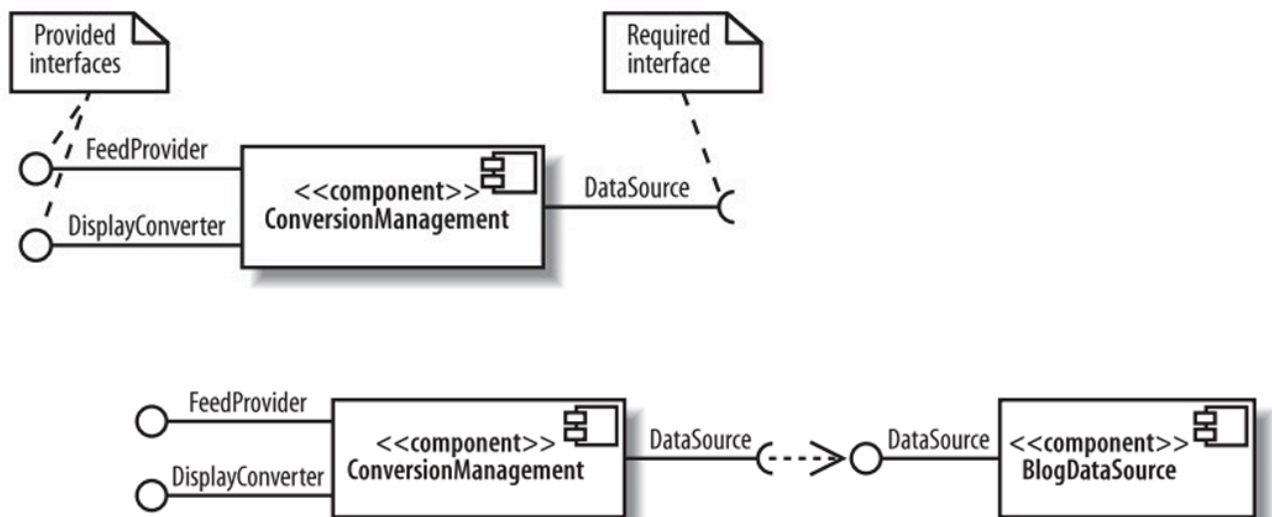


Figura 1

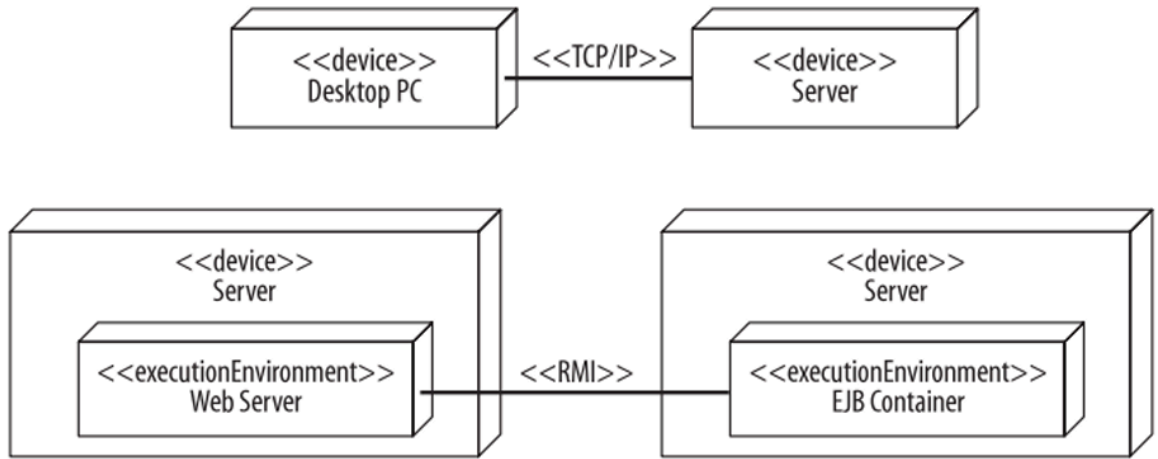


Figura 2

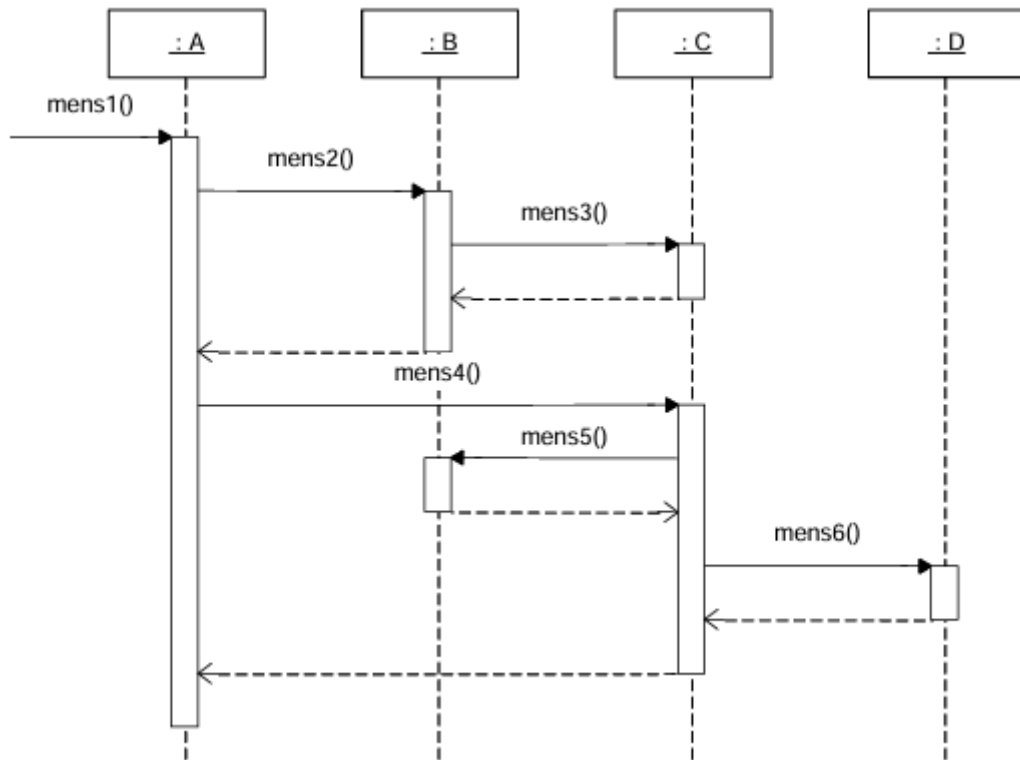


Figura 3

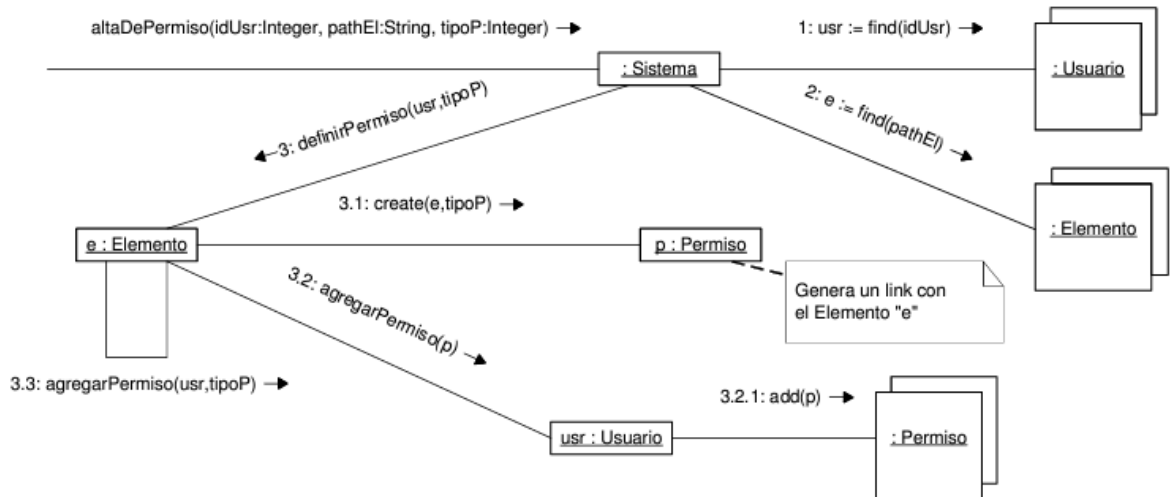


Figura 4

- a. Sólo Figura 1
- b. Sólo Figura 2
- c. Sólo Figura 3 y Figura 4
- d. Ninguna figura podría representar un diagrama de componentes.

18. En una arquitectura cliente-servidor la funcionalidad del sistema se organiza en servicios, y cada servicio lo entrega un servidor independiente. Los clientes son usuarios de dichos servicios. Además, esta arquitectura:

- I. No es una arquitectura muy utilizada ya que en la actualidad se busca diseñar un patrón arquitectónico que responda a arquitecturas de microservicios.
 - II. Los servidores se pueden distribuir a través de una red. La funcionalidad general estaría disponible a todos los clientes, no necesita implementarse en todos los servicios.
 - III. No es recomendable que la combinemos con otros tipos de patrones arquitectónicos ya que el mantenimiento puede ser bastante complejo.
 - IV. Cada servicio es un punto de falla. El rendimiento depende de la red, así como del sistema. Pueden existir problemas administrativos cuando los servidores son propiedad de diferentes organizaciones.
- a) Solo las afirmaciones II, III y IV son correctas.
 - b) Solo las afirmaciones II y IV son correctas.
 - c) Solo la afirmación II es correcta.
 - d) Solo las afirmaciones I, II y III son correctas.

19. Acerca de los patrones de Diseño de software ¿qué afirmación o afirmaciones son correctas?:

- I. Permiten seleccionar una solución entendible y probada a ciertos problemas, definiendo los principios organizativos del sistema.
 - II. Al basar la arquitectura en patrones que son conocidos se facilita comunicar las características importantes de la misma.
 - III. El patrón se presenta en una forma estándar que permite que sea fácilmente reutilizado.
 - IV. No es necesario conocer ciertos patrones arquitectónicos porque en los equipos de desarrollo siempre hay alguien con experiencia que conoce la tendencia del mercado y es recomendable seguir su sugerencia.
- a. Solo las afirmaciones II, III son correctas.
 - b. Solo las afirmaciones II y IV son correctas.
 - c. Todas las afirmaciones son correctas.
 - d. Solo las afirmaciones I, II y III son correctas.

20. Indique cuáles de las siguientes afirmaciones sobre la reingeniería y la refactorización en el desarrollo de software son correctas.

I) La refactorización implica mejorar un programa para mejorar su estructura, reducir su complejidad o hacerlo más fácil de entender.

II) La reingeniería puede suceder luego que el sistema está en etapa de mantenimiento y los costos se incrementan.

III) En la reingeniería debe participar el cliente, dado que es común que se modifiquen o se agreguen nuevas funcionalidades.

IV) La reingeniería carece de riesgos, dado que implica realizar modificaciones de código sin cambiar las funcionalidades.

- a) Sólo las afirmaciones I y III son correctas
- b) Sólo las afirmaciones II y IV son correctas
- c) Sólo las afirmaciones III y IV son correctas
- d) Sólo las afirmaciones I y II son correctas

21. ¿Cuál de las siguientes opciones describe de forma más adecuada por qué los sistemas de software necesitan evolucionar?

- a) Porque el hardware siempre es más caro que el software, y actualizarlo reduce costos.
- b) Porque los sistemas de software necesitan adaptarse al entorno operativo, mejorar en funcionalidades y corregir errores.
- c) Porque todos los sistemas eventualmente fallan y deben ser reemplazados en su totalidad.
- d) Porque la evolución elimina la necesidad de pruebas de regresión y documentación.