

Instrucciones:

- Cada pregunta múltiple opción contestada correctamente tiene un valor de 3,34 puntos. Cada pregunta incorrecta de la múltiple opción resta 1,11 puntos.
- **El puntaje total del examen es 100 puntos y se aprueba con 60 o más puntos.**

Múltiple Opción

1. En relación a la verificación y validación:
 - a) El testing es la actividad de localizar el defecto en el código y corregirlo.
 - b) (a) y la verificación busca comprobar que el sistema cumple con los requerimientos especificados.
 - c) Categorizar y registrar los tipos de defectos puede servir para mejorar el proceso y como guía para orientar la verificación.
 - d) (c) y uno de los objetivos de la verificación y validación es ejecutar el programa para medir su calidad.
2. En el contexto del desarrollo ágil y TDD:
 - a) En TDD, tal como lo indican sus siglas en inglés, las pruebas permiten guiar el desarrollo de software. A partir de las historias de usuario se construyen las pruebas unitarias. No se comienza a implementar hasta que no estén las pruebas diseñadas y el código no está listo para integrarse hasta que no pasa todas las pruebas.
 - b) En esta técnica es necesario que el código se construya antes del diseño de las pruebas. Además, se debe tener en cuenta que las pruebas sirvan para verificar los criterios de aceptación de la historia de usuario.
 - c) Ninguna afirmación es correcta ya que TDD no es una técnica de desarrollo dirigido por pruebas.
 - d) Una ventaja de esta técnica es que las pruebas se diseñan a partir de las historias de usuario y, por su parte, las historias de usuario están siempre bien definidas y detalladas por los interesados. Por esto, los casos de prueba serán muy efectivos.
3. Las técnicas de verificación estática:
 - I. Analizan el producto para deducir su correcta operación.
 - II. Se pueden usar tanto para verificación como para validación.
 - III. Sirven para verificar requerimientos, diseño, código y casos de prueba.
 - IV. Normalmente permiten detectar un único error por prueba ya que hay errores que cubren a otros errores.
 - V. Las inspecciones son una técnica de verificación estática.

Seleccione la opción correcta:

- a) Todas son correctas.
 - b) Sólo (I), (III) y (V) son correctas.
 - c) Sólo (I), (II), (III) y (V) son correctas.
 - d) Sólo (II), (V) son correctas.
4. Los casos de prueba se componen de:
 - I. Objeto de prueba.
 - II. Entradas requeridas y salidas esperadas.
 - III. Defectos encontrados.
 - IV. Precondiciones (contexto en el cual debe ejecutarse).
 - V. Pasos o procedimiento de ejecución.
 - VI. Artefactos de software o documentación a los cuales está relacionado (especificación de requisitos, diseño, arquitectura, etc).

Seleccione la opción correcta:

- a) Todas las opciones son correctas.
 - b) Sólo 5 opciones son correctas.
 - c) Sólo (II), (III), (IV) y (V) son correctas.
 - d) Sólo (I), (II) y (IV) son correctas.
5. En lo que respecta al Diseño
 - a) Es una actividades básicamente creativa.
 - b) (a) donde buscar reducir la complejidad es esencial para su entendimiento.
 - c) Debe seguir los pasos recomendados en la bibliografía, en particular el uso de patrones de diseño.
 - d) (c) así como atender los aspectos de verificación que se apliquen a los requisitos no funcionales.

6. La planificación de la verificación y validación en un proyecto de software debe considerar los siguientes aspectos:
- Un error común al elaborar el plan es considerar como recursos a los desarrolladores. Las pruebas no deben involucrar a todo el equipo, solamente a los testers, el líder del QA y el arquitecto (para validar el diseño).
 - Cuando se está definiendo la estrategia, se deben tener en cuenta los siguientes niveles: nivel unitario, nivel de integración, nivel de sistema y nivel de aceptación.
 - Para las pruebas de integración debe asegurarse la disponibilidad del cliente para que valide que el sistema cumpla con sus necesidades.
 - Para las pruebas de integración se debe tener en cuenta el costo del entorno de pruebas. Sin embargo, la estrategia no puede considerar el tiempo que puede insumir preparar el entorno, ya que depende del tamaño del software a desarrollar y su complejidad.
7. En relación a los modelos de procesos:
- La aplicación del modelo en cascada puede presentar dificultades para responder a los cambios de requisitos.
 - Uno de los beneficios de scrum, en relación al modelo de cascada, es que se pueden realizar entregas más rápidas de software que pueden ser útiles para el cliente.
 - Con el enfoque de integración y configuración se reducen algunos costos y riesgos ya que menos software se desarrolla desde cero.
 - Las actividades de especificación, desarrollo, validación y evolución se pueden organizar de diferente forma de acuerdo al modelo de proceso a utilizar.
 - Los sistemas de software pueden ser desarrollados incorporando ideas y elementos de distintos modelos de procesos de desarrollo.

Seleccione la opción correcta:

- Todas son correctas.
- Sólo (I), (III), (IV) y (V) son correctas.
- Sólo (I), (II), (IV) y (V) son correctas.
- Sólo (II), (III), (IV) y (V) son correctas.

8. Con respecto a los métodos ágiles:
- Entre otras cosas, los métodos ágiles buscan evitar generar documentación que luego no se consulte o actualice.
 - Las metodologías ágiles esperan que los requisitos cambien. Por tal motivo es una buena práctica diseñar el sistema para que se adapte a los cambios.
 - Los procesos ágiles necesitan ser gestionados al igual que los basados en planes (aunque en general de forma diferente).
 - Las historias de usuario se pueden utilizar como base para la planificación de las iteraciones del proyecto.
 - Scrum indica que debe utilizarse junto con las técnicas de programación por pares y desarrollo guiado por las pruebas.

Seleccione la opción correcta:

- Todas son correctas.
- Sólo (I), (II), (III) y (IV) son correctas.
- Sólo tres opciones son correctas.
- Sólo (II), (III), (IV) y (V) son correctas.

9. Teniendo en cuenta las actividades básicas que se realizan en los procesos de software:
- En los métodos ágiles el diseño de software es una actividad separada de la construcción de software y se realizan en momentos diferentes.
 - El diseño de la arquitectura y de la base de datos son actividades que suelen formar parte del proceso de diseño.
 - En los métodos ágiles no se realizan actividades de diseño, ya que se hace refactoring cuando es necesario y esto se hace directamente a la hora de codificar.
 - Las actividades que se realizan en los procesos de software se organizan diferente de acuerdo al modelo de proceso de desarrollo usado.
 - Siempre se utilizan historias de usuario para especificar los requisitos en metodologías ágiles.

Seleccione la opción correcta:

- Solo tres son correctas.
- Sólo una opción es correcta.
- Sólo (II) y (IV) son correctas.
- Sólo (I), (II), (IV) y (V) son correctas.

10. El cambio es inevitable en todos los grandes proyectos de software. En relación a esto:
- Los cambios en el negocio pueden conducir a cambios en los requisitos de los sistemas.
 - Los cambios por lo general implican retrabajo. El costo de un cambio implica tanto el costo de re-analizar los requisitos como el costo de la implementación de esa nueva funcionalidad.
 - Para reducir el costo del retrabajo se pueden incluir actividades que puedan anticipar los posibles cambios. Un ejemplo de esto es desarrollar un prototipo para validar las funcionalidades claves de un sistema.
 - Un enfoque posible para reducir los costos del retrabajo es que los procesos de software sean diseñados para que los cambios se puedan introducir de forma sencilla en el sistema. El modelo en V se creó con este objetivo.
 - Un prototipo puede ser utilizado para demostrar conceptos, probar opciones de diseño y validar requisitos.

Seleccione la opción correcta:

- Todas son correctas.
 - Sólo (I), (III) y (IV) son correctas.
 - Sólo (I), (II), (III) y (V) son correctas.
 - Sólo (II), (III), (IV) y (V) son correctas.
11. En relación a escalar los métodos ágiles:
- Los métodos ágiles no se pueden escalar.
 - Es importante considerar las características del sistema a construir, del equipo de desarrollo, de la organización que desarrolla el producto y de la organización del cliente.
 - Para poder escalar se debe tener en cuenta que es necesario contar con una planificación más estricta y que las liberaciones deben ser menos frecuentes.
 - “Escalar hacia arriba” refiere a cómo los métodos ágiles pueden ser introducidos a lo largo de una gran organización con muchos años de experiencia en el desarrollo de software.
12. Respecto a la gestión de proyectos. Seleccione la opción **incorrecta**:
- La gestión de proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades del proyecto para cumplir con sus requisitos.
 - La incertidumbre al planificar un proyecto varía dependiendo de la etapa en la que se encuentre.
 - El alcance del producto es lo mismo que el alcance del proyecto.
 - Para crear una estructura de desglose de trabajo (EDT) es necesario subdividir los entregables del proyecto y el trabajo en componentes más pequeños.
13. Seleccione la opción correcta:
- Si ante un conflicto el director de proyectos dice “Este tema está tardando más de lo esperado en resolverse y no han llegado a un acuerdo, vamos a ir por la opción presentada por Juan”, la estrategia aplicada es suavizar/ adaptarse.
 - Si ante un conflicto el director de proyectos dice “Este tema está tardando más de lo esperado en resolverse y no han llegado a un acuerdo, vamos a retomarlo en la próxima reunión”, la estrategia aplicada es suavizar/adaptarse.
 - La teoría de X y la Y presenta dos enfoques de la naturaleza humana. En la X se afirma que a las personas no les gusta el trabajo, que no tienen ambición, etc. En la Y que si son motivados cumplen altas expectativas, son autodisciplinados, etc.
 - La teoría de Maslow presenta 5 niveles donde el más alto corresponde a las personas con mayores ingresos económicos. Si no se cuenta, con ingresos altos, no se puede llegar a ese nivel.
14. Respecto a la estimación de un proyecto:
- Todas las técnicas de estimación para proyectos de software incorporan, de diferentes maneras y en diversos grados, experiencias pasadas, conocimiento del futuro y factores de ajuste.
 - Periódicamente, cuando se realice el seguimiento del proyecto se evalúa si es necesario o no re-estimar.
 - Es necesario re-estimar un proyecto si en su desarrollo se registran problemas en alguna de las tecnologías a utilizar.
 - El tipo de software a construir no es un factor que influya en la estimación de un proyecto.

Seleccione la opción correcta:

- Todas las opciones son correctas.
- Sólo (I), (II) y (IV).
- Sólo (I), (II) y (III).
- Sólo (III) y (IV).

15. Respecto al desarrollo y seguimiento del cronograma de un proyecto. Seleccione la opción correcta:
- Las actividades mínimas para generar el cronograma son: identificar y estimar las tareas, que generalmente se desprenden de los paquetes de trabajo del EDT, y asignar las tareas al equipo de desarrollo.
 - Siempre es posible adelantar la fecha de entrega de un proyecto sin aumentar costos ni riesgos del proyecto.
 - Las actividades necesarias para generar el cronograma son independientes entre sí, esto implica que una actividad no tiene incidencia sobre otra.
 - Aunque el proyecto cuente con un $SPI < 1$ no se puede afirmar que esté por encima o por debajo del presupuesto.
16. Dado un proyecto que consiste en desarrollar un sistema de inventario para el cual el cliente definió que se debe utilizar el sistema de autenticación que ya utilizan en su organización. Este sistema de autenticación cuenta con muy poca documentación y se dice que es poco robusto. De acuerdo a las siguientes afirmaciones:
- En la gestión del proyecto se puede asumir que no se va a tener ningún problema respecto a la integración con el sistema de autenticación.
 - Que exista algún problema en la integración con el sistema de autenticación es un riesgo negativo que se debe gestionar.
 - Que exista algún problema en la integración con el sistema de autenticación es un riesgo negativo que se debería aceptar, dado que no existe la posibilidad de reducir la probabilidad ni el impacto del riesgo.
 - Una forma de mitigar el riesgo es realizar prototipos al inicio del proyecto para integrarse con el sistema de autenticación y evaluar el comportamiento.
 - Una forma de evitar el riesgo podría ser proponerle al cliente mejorar el sistema de autenticación para que sea más robusto.
- Seleccione la opción correcta:
- Sólo dos afirmaciones son correctas.
 - Sólo tres afirmaciones son correctas.
 - Sólo cuatro afirmaciones son correctas.
 - Todas las afirmaciones son correctas.
17. En lo visto en el curso sobre los productos genéricos y los productos específicos de software:
- Los productos genéricos corresponden a sistemas de software que se comercializan y se venden a cualquier cliente que desee comprarlos.
 - Los productos personalizados corresponden a software que es encargado por un cliente específico para satisfacer sus propias necesidades.
 - La distinción entre productos genéricos y productos específicos radica en quién paga por el producto.
 - Dos de las anteriores son correctas.
18. Respecto a la especificación de requisitos, los requisitos pueden ser especificados desde la perspectiva del sistema o desde la perspectiva del usuario.
- Debe elegirse sólo uno de estos estilos para que la especificación resulte coherente.
 - Es correcto mezclar estos estilos, expresando cada requisito desde la perspectiva que aporte mayor claridad a la especificación.
 - Cuando se toma la decisión de especificar los requisitos desde el punto de vista del usuario, la mejor práctica es considerar al usuario genérico y no a un perfil de usuario en particular, ya que esto último hace que la especificación quede demasiado rígida.
 - Si bien existen estos dos estilos de especificación de requisitos, es indistinto expresarlos desde una u otra perspectiva, ya que no aporta a la comprensión de los requisitos sino que depende de lo que resulte más cómodo para quien está a cargo de la especificación.
19. Respecto a los requisitos de un sistema, seleccione la opción correcta:
- Un atributo de calidad es un tipo de requisito no funcional que describe una característica de servicio o desempeño de un producto.
 - Un atributo de calidad es un tipo de requisito funcional que describe lo que el sistema debe hacer bajo condiciones específicas.
 - Los requisitos del dominio se derivan de las necesidades específicas de los usuarios.
 - Los requisitos funcionales describen cómo el sistema debería reaccionar a entradas particulares y cómo debería comportarse en situaciones particulares, pero no describen lo que el sistema no debe hacer.

20. Respecto a la trazabilidad de los requisitos:
- Es necesario realizar un análisis de costo-beneficio para decidir con qué otros artefactos de desarrollo serán trazables los requisitos, teniendo en cuenta el uso que se dará a cada trazabilidad.
 - Uno de los principales beneficios de mantener la trazabilidad de los requisitos con artefactos de diseño y código, es facilitar el análisis del impacto de un cambio en el sistema.
 - La trazabilidad debe establecerse desde los requisitos hacia artefactos de diseño, implementación y testing, ya que es de mucha utilidad para asegurar la mantenibilidad del sistema. Sin embargo, no resulta útil mantener la trazabilidad desde los requisitos hacia su origen, es decir hacia las necesidades del negocio.
 - Una buena gestión de la configuración facilita la tarea de mantener la trazabilidad entre diferentes artefactos.
 - La trazabilidad desde los requisitos hacia las necesidades del negocio ayuda a identificar requisitos que no son necesarios.

Seleccione la opción correcta:

- Solo (I), (II), (III) y (IV) son correctas.
 - Solo (I), (II), (IV) y (V) son correctas.
 - Solo (II), (IV) y (V) son correctas.
 - Solo (II) y (IV) son correctas.
21. Respecto a la obtención de los requisitos, cuando existe un sistema anterior una de las técnicas de obtención de requisitos es la observación de su interfaz de usuario
- Es una técnica muy útil ya que permite obtener cuáles serán todas las funcionalidades del nuevo sistema.
 - Ayuda a determinar cuál será el diseño de la interfaz del nuevo sistema, ya que no es conveniente que ésta cambie demasiado para evitar la resistencia de los usuarios a utilizar el nuevo sistema.
 - Esta técnica es un insumo valioso para elaborar los casos de uso del nuevo sistema.
 - Garantiza ahorrar tiempo de entrevistas con los principales involucrados.
22. Respecto a los casos de uso:
- Un caso de uso expandido refiere a un caso de uso que se relaciona con otro a través de la relación de extensión.
 - Es necesario elaborar casos de uso para todos los requisitos del sistema, tanto funcionales como no funcionales.
 - Los casos de uso son independientes de la metodología de diseño e implementación del sistema.
 - Los diagramas de casos de uso sirven para visualizar todos los detalles de la interacción entre un actor y el sistema, como por ejemplo el diseño de los campos de las pantallas que tendrá el sistema.
 - En un diagrama de caso de uso es importante especificar el rol de cada actor que interactúa con el sistema.

Seleccione la opción correcta:

- Solo (I), (III), (IV) y (V) son correctas.
 - Solo (III), (IV) y (V) son correctas.
 - Solo (III) y (V) son correctas.
 - Todas las opciones son correctas.
23. Los conceptos claves para un buen diseño son que
- Sea correcto.
 - Busque alta cohesión y acoplamiento en sus componentes.
 - (a) y evitar así que las restricciones del proyecto afecten el cronograma.
 - Ninguna de las anteriores es correcta.
24. Si su rol es ser Arquitecto de Software en un proyecto de mediano porte:
- Deberá estar atento a las restricciones del proyecto y cómo pueden afectar las pautas de diseño.
 - Deberá entender en detalle todos los casos de uso, ya que son fundamentales para la arquitectura.
 - Trabjará directamente en la estimación de requisitos, ya que el esfuerzo de cada requisito debe considerarse al diseñar la arquitectura.
 - (c). Deberá estimar, en conjunto con el resto del equipo de proyecto, el esfuerzo de implementación del producto.
25. Un diseño debe ser:
- Verificable.
 - Complejo.
 - Verdadero.
 - (a) y (c) son correctas.

26. Sobre la construcción de software:
- Refiere a la creación detallada del software.
 - (a) que combina codificación, verificación.
 - (b) así como pruebas unitarias, de integración y debugging.
 - Busca evitar el cambio, ya que este genera retrabajo y costos.
27. En un proceso normal de evolución de software:
- La actividad de implementación del cambio implica un análisis de los requisitos de cambio del cual pueden surgir modificaciones a la propuesta de cambio.
 - La implementación del cambio es una actividad muy importante, sin embargo el análisis de impacto y la planificación de la nueva versión son importantes únicamente en sistemas de gran porte o cuando el equipo encargado de implementar el cambio es diferente al que desarrolló el sistema.
 - La evolución de un sistema que está en producción siempre está motivada por un conjunto de requisitos que provienen de actores del negocio.
 - En la actividad de planificación de la nueva versión del sistema, solo deben incluirse los requisitos de cambio que representan mayor riesgo para el negocio, a los efectos de realizar un mejor seguimiento de la implementación de los mismos.
28. En un proceso de evolución de software de emergencia:
- El cambio puede ser implementado sin pasar por todas las etapas del proceso de ingeniería de software únicamente si el equipo encargado de implementar el cambio es el mismo que desarrolló el sistema.
 - Si un cambio de emergencia fue implementado sin pasar por todas las etapas del proceso de ingeniería de software, entonces la mejor práctica es no cambiar esa implementación ya que esto insumiría un gran esfuerzo en pruebas de regresión del equipo de desarrollo.
 - En la implementación de un cambio de emergencia, el mayor riesgo es que la documentación de los requisitos y el diseño quede desactualizada, impactando en la mantenibilidad del sistema.
 - Una vez implementado y liberado un cambio de emergencia, debería agregarse una petición de cambio para realizar los ajustes que surjan de la aplicación del proceso formal de ingeniería de software.
 - Dado que la implementación de un cambio de emergencia implica modificar el código directamente, en general no se obtiene la mejor solución de implementación al cambio propuesto.

Seleccione la opción correcta:

- Solo (III), (IV) y (V) son correctas.
 - Solo (I), (III), (IV) y (V) son correctas.
 - Solo (IV) y (V) son correctas.
 - Ninguna de las opciones es correcta.
29. Acerca del mantenimiento del software se conoce que:
- El mayor porcentaje del esfuerzo destinado al mantenimiento lo consume el mantenimiento correctivo o reparación de fallas.
 - Siempre es mejor agregar funcionalidades luego de haber construido el software en lugar de desarrollar la misma funcionalidad al comienzo del desarrollo, ya que se reutilizan los componentes construidos.
 - Cuando el equipo de mantenimiento es diferente al equipo de desarrollo cada vez que se tiene que implementar un cambio en un componente que no fue modificado hasta al momento, debe realizarse una comprensión del programa.
 - Para las pruebas de integración se debe tener en cuenta el costo del entorno de pruebas. Sin embargo, la estrategia no puede considerar el tiempo que puede insumir ya que esto depende únicamente del tamaño del software a desarrollar.
30. Con respecto a la evolución de un sistema cuyo desarrollo inicial se realizó siguiendo un modelo ágil:
- Para la implementación de un cambio no se recomienda aplicar prácticas de desarrollo ágil ya que la dinámica de iteraciones para obtener incrementos de producto se interrumpió al liberar el sistema en producción.
 - Para la implementación de un cambio se recomienda aplicar prácticas de desarrollo ágil únicamente si el equipo encargado de implementar el cambio está familiarizado con los métodos ágiles.
 - Si el equipo encargado de la implementación del cambio normalmente trabaja siguiendo un enfoque tradicional, entonces el problema más común en la transferencia entre ambos equipos es que el código de la aplicación no ha sido simplificado ni refactorizado.
 - Las pruebas de regresión automatizadas son muy útiles para el equipo encargado de la implementación del cambio.